

Zpracování dat a jejich využití pro modelování a simulaci dopravy

Data Processing and its Utilization for Traffic Modelling and Simulation

Zadání diplomové práce

Student: **Bc. Libor Vymětalík**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Zpracování dat a jejich využití pro modelování a simulaci dopravy**
Data Processing and its Utilization for Traffic Modelling and Simulation

Zásady pro vypracování:

Cílem práce je analýza dopravních dat a jejich následné zpracování s využitím statistických, analytických a dalších nástrojů.

1. Zmapujte oblast dopravních dat a možnost přístupu k těmto datům, především z pohledu reálných dat o dopravních tocích a situacích v reálném čase.
2. Analyzujte vztahy mezi jednotlivými typy dat a navrhnete komplexní datový model ve vazbě na infrastrukturu.
3. Navrhnete komplexní přístup pro vytvoření popisu veličin dopravní infrastruktury na základě zpracování dopravních dat a jejich statistického vyhodnocení.
4. Klíčové prvky navrženého přístupu ukázkově implementujte a vizualizujte.
5. Zhodnot'te možnosti použití vybraných dat a jejich zpracování.

Seznam doporučené odborné literatury:

- [1] F. Mannering: Principles of Highway Engineering and Traffic Analysis, ISBN: 0470290757, Wiley, 2008
- [2] S. Blatnig: Microscopic Traffic Simulation with Intelligent Agents: Simulation of Human Driving Behaviour, ISBN: 978-3639224153, VDM Verlag Dr. Müller, 2009
- [3] www.jsdi.cz - NDIC

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Michal Radecký, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2015




doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty


Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostravě 28. dubna 2015


.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2015


.....

Chtěl bych poděkovat vedoucímu diplomové práce **Ing. Michalu Radeckému, Ph.D.** za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Abstrakt

Tato diplomová práce je zaměřena na získávání a zpracování dopravních informací. Dále jsou v rámci práce stručně popsány systémy, které tato data poskytují a jaké jsou možnosti lokalizace výskytu události. Součástí je i návrh modelu, který by umožňoval vkládání několika typů dopravních informací z několika různých zdrojů. Tento model byl poté zjednodušen pro potřeby implementace, která se skládá ze dvou samostatných celků. První celek získává a zpracovává data, druhý celek tato data zobrazuje pomocí webové aplikace. V poslední části práce jsou nad těmito daty provedeny analýzy, které zjišťují, jaké množství informací přichází na jednotlivé části naší republiky. Současně jsou zobrazeny i statistické informace.

Klíčová slova: analýza dat, dopravní data, MSSQL, telematické systémy, Jednotný systém dopravních informací

Abstract

My thesis is focused on getting and processing the travel information. I briefly describe the systems which provide these data and the possibilities of locating occurrence of events. The part of thesis is also a suggestion of model which would be able to insert a various kinds of travel information from few different sources. This model was simplified for needs of implementation which consists of two individual units. The first one is dealing with and acquiring data, the second unit displays data with help of web app. In the final analysis, I had made a calculation of data. It does make sure that the right amount of information comes to different parts of our country. At the same time, you can also find statistical information.

Keywords: data analysis, data transport, MSSQL, telematics systems, Unified Traffic Information System

Seznam použitých zkratk a symbolů

AJAX	– Asynchronous JavaScript and XML
API	– Application programming interface
C#	– C Sharp
DDR	– Distribuční datové rozhraní
DI	– Dependency Injection
FCD	– Float car data
FTP	– File Transfer Protocol
GPS	– Global Positioning System
HTTP	– Hypertext Transfer Protocol
JSDI	– Jednotný systém dopravních informací
JSON	– JavaScript Object Notation
MAC	– Media Access Control
MSSQL	– Microsoft SQL Server
MVC	– Model-view-controller
MVP	– Model-view-presenter
NDIC	– Národní dopravní informační centrum
PHP	– Hypertext Preprocessor
RDS	– Radio Data System
RSS	– Really Simple Syndication
ŘSD	– Ředitelství silnic a dálnic
S-JTSK	– Systém jednotné trigonometrické sítě katastrální
SMTP	– Simple Mail Transfer Protocol
SQL	– Structured Query Language
SÚS	– Správa a údržba silnic
TMC	– Traffic Message Channel
UTM	– Universal Transverse Mercator
WGS-84	– World Geodetic System
XML	– Extensible Markup Language

Obsah

1	Úvod	6
1.1	Techniky sběru dat	6
1.2	Využití dat	7
1.3	Šíření dopravních dat	8
2	Systémy dopravních dat	10
2.1	Jednotný systém dopravních informací	10
2.2	ViaRodos.cz	13
2.3	CzechTraffic.cz	14
2.4	Lokální dopravní centra	14
3	Možnosti lokalizace události	16
3.1	Lokační tabulky	16
3.2	Lokační tabulky pozemních komunikací	17
3.3	Souřadnicové systémy	18
4	Návrh modelu	21
4.1	Obecný návrh modelu	21
4.2	Realizovaný datový model	27
5	Získávání dat z JSDI	29
5.1	Příjem dat	29
5.2	Validace dat	30
5.3	Výpadky komunikace	30
5.4	Zvolení technologie pro přenos	30
5.5	Zjištěné problémy	30
6	Implementace systému	32
6.1	Procesy v aplikaci	32
6.2	Získávání dopravních informací	35
6.3	Webová vrstva	41
6.4	Webová aplikace	44
7	Analýza a statistiky	49
7.1	Analýza četnosti událostí	49
7.2	Statistiky dopravních informací	54
8	Závěr	58
9	Reference	60
	Přílohy	62

A Přílohy	63
A.1 Struktura DVD s přílohou	63
A.2 Testovací sestava	63
B Tabulky	64
B.1 Datový slovník	64
B.2 Lineární schéma závislostí	67
C Obrázky	68
D Zdrojové kódy	69
D.1 Zdrojový kód kontroly mezipaměti	69
D.2 Zdrojový kód výpočtu základu logaritmu	69

Seznam tabulek

1	Typy pozemních komunikací [26]	21
2	Entita AdministrativeStreetUnit	64
3	Entita AdministrativeUnit	64
4	Entita AffectedRoadPart	64
5	Entita Country	64
6	Entita EventsCode	64
7	Entita Evi	65
8	Entita Region	65
9	Entita RoadInfo	65
10	Entita RoadClassCode	65
11	Entita StringId	65
12	Entita TrafficAnalyticsCache	65
13	Entita TrafficAnalyticsCacheItems	66
14	Entita TrafficData	66
15	Entita TrafficType	66
16	Entita UpdateClassCode	66
17	Entita VersionData	66

Seznam obrázků

1	Funkce mýtné brány [2]	7
2	Zobrazení některých telematických systémů a systémů pro šíření dat [10]	9
3	Schéma získávání dat JSDI [11]	10
4	Okno aplikace Viarodos.cz	13
5	Zdroje dat a následné šíření u systému CzechTraffic.cz [15]	14
6	Okno aplikace DopravniDostupnost.cz	15
7	Schéma WGS-84 zobrazení (zdroj: http://gis.zcu.cz)	18
8	Schéma Křovákova zobrazení [25]	20
9	Kamerový snímek z Ostravy (zdroj: http://mapy.ovanet.cz/krizovatky)	24
10	ER diagram obecného modelu	25
11	Vkládání dat z různých systémů	26
12	ER Diagram realizovaného modelu	27
13	Diagram aktivity získávání a ukládání dat	32
14	Diagram aktivity automatické kontroly cizích klíčů	33
15	Diagram aktualizace dat	34
16	Tok dat u parsovací části	35
17	Okno se získanými informacemi	39
18	Tok dat u webové části	41
19	Ukázka zobrazení webové stránky	45
20	Výběr typu události	45
21	Výběr události podle typu pozemní komunikace	46
22	Zvýraznění kraje	46
23	Výběr události podle data	47
24	Analýza událostí	47
25	Porovnání zobrazení detailu události	48
26	Zobrazení verzí události	48
27	Četnost událostí pro všechny typy s velikostí čtverce 5 km	51
28	Četnost nehod s velikostí čtverce 5 km	52
29	Četnost nehod v Ostravě, čtverec 3 km	52
30	Četnost nebezpečných situací s velikostí čtverce 5 km	53
31	Počet nehod podle dnů v týdnu	54
32	Rozdělení nehod podle hodin	55
33	Událostí podle dnů	55
34	Počet verzí dle dnů	56
35	Události podle typu	56
36	Události podle kraje	57
37	Zobrazení okna s výpisem chyb	68
38	Prohlížeč událostí	68

Seznam výpisů zdrojového kódu

1	Zobrazení výstřižku XML zprávy s chybným číslováním verzí	31
2	Spuštění několika služeb současně	40
3	Ukázka zaslání payloadu z presenteru	43
4	Automatická invalidace cache při změně souboru	44
5	Ukládání počtu záznamů pro jednotlivé tabulky	69
6	Výpočet základu logaritmu	69

1 Úvod

Dopravní vozidla jsou každodenním dopravním prostředkem pro miliony lidí po celém světě. Ať už lidé cestují do práce, školy či za jiným účelem, vždy chtějí do cíle cesty dorazit v bezpečí a bez zbytečných komplikací. To by však s absencí dopravních informací bylo prakticky nemožné. Tato diplomová práce pojednává právě o možnostech zpracování dopravních dat, které mohou cestování velmi zpříjemnit.

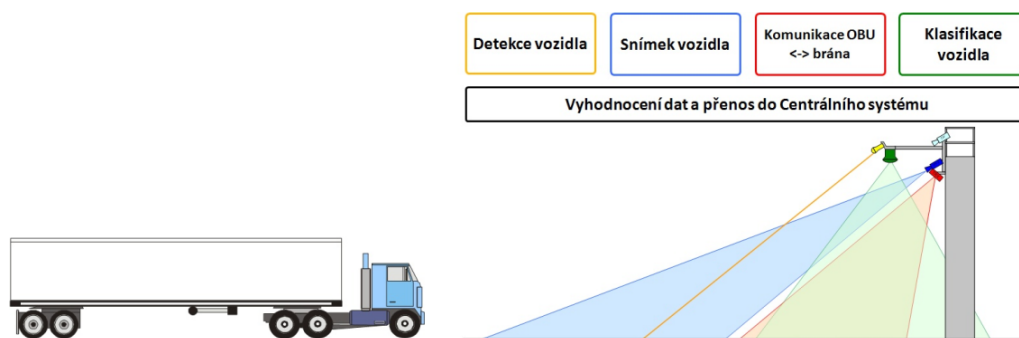
Cílem této práce je tedy seznámení s možnostmi využití a zpracování dopravních informací. V úvodu práce je stručně nastíněno, proč je důležité dopravní data získávat a dále s nimi pracovat. V dalších částech jsou zmíněny některé systémy, které tyto data poskytují a možnosti jak lze vyskyt těchto události lokalizovat.

1.1 Techniky sběru dat

Častým problémem velkých měst a některých silnic jsou dopravní kolapsy způsobené vysokou intenzitou dopravy. Intenzita dopravy na pozemní komunikaci zjednodušeně znamená, jak moc je tato komunikace využívána. Obecně lze určit, že čím vyšší je intenzita, tím je nižší průměrná rychlost vozidel. Pro dlouhodobější výpočty intenzity a modelování dopravy je důležitá tyto data nějakým způsobem získávat. Pro tyto účely existují dvě možnosti. První možností je ruční sběr informací. Sčítání vozidel tak na určeném úseku provádí pověřený pracovník s patřičným formulářem, do kterého pouze zapisuje informace o počtech a typech vozidel. Tato metoda sběru však není příliš efektivní. Další možnosti, jak tyto data získávat jsou již zcela automatické. Menší nevýhodou však může být, že některé typy automatických měřičů nedokáží určit, o jaký typ vozidla jde. Mezi nejčastější technologie pro sběr dat patří.

- **Obrazové analýzy** – technologie založená na analýze obrazu z dopravních kamer. Údaje mohou být vkládány pomocí automatické analýzy, případně manuálně operátorem. Automatická analýzy počítání vozidel je výborně vyřešena na křižovatkách v Ostravě.
- **Indukční smyčky** – na některých silnicích a dálnicích je v povrchu vozovky zabudovaná speciální indukční smyčka. Ta je tvořena oscilátorem, který má nastavenou určitou konstantní frekvenci. Při průjezdu vozidla dojde ke změně indukce a tím dojde ke změně frekvence na oscilátoru. Pokud tato změna dosáhne nadefinované hodnoty, je vnímána jako průjezd vozidla. Pomocí této techniky lze tedy určit, kolik vozidel daným úsekem projede v určitém časovém intervalu. [1]
- **Mýtné brány** – mýtné brány slouží především pro výběr poplatků na zpoplatněných úsecích dálnic a silnic. Jedná se však také o speciální telematický systém, který pomocí mikrovln dokáže komunikovat s jednotkou umístěnou za čelním sklem vozidla. Pomocí této komunikace lze spočítat, kolik vozidel pod mýtnou bránou za určitý interval projelo. Nevýhodou však je, že povinnost umístění této jednotky za oknem vozidla se vztahuje pouze pro vozidla nad 3,5 tuny [2]. Na obrázku 1 jsou zobrazeny funkce mýtné brány.

- **GPS FCD** – tento druh sběru dat o dopravní intenzitě je založen na velkém množství automobilů, které jsou vybaveny speciální jednotkou s GPS přijímačem. Tato jednotka je aktivována automaticky při nastartování vozidla. Během jízdy jednotka shromažďuje data jako GPS polohu, směr jízdy, rychlost, atp. Tyto data jsou v určitém intervalu, který je závislý na rychlosti pohybu vozidla, automaticky zasílána na přijímací server. Ten data zpracuje a uloží. [3]
- **Bluetooth FCD** – v některých městech funguje podobný princip jak výše zmíněné GPS FCD. Jedinou odlišností je technologie sběru dat. Tento princip je založen na sběru dat pomocí technologie Bluetooth. Každé zařízení s touto technologií obsahuje unikátní MAC adresu, která umožňuje toto zařízení jednoznačně identifikovat. Po přiblížení k jednotce (nejčastěji na křižovatkách ulic), která je zodpovědná za hledání Bluetooth zařízení je tato MAC adresa získána a uložena. Jak vozidlo pokračuje dále po komunikaci, dorazí na další křižovatku. Zde je opět MAC adresa zařízení přečtena a uložena. Systém je podle těchto dat poté schopen dopočítat, jak dlouho trval průjezd tohoto segmentu. [4] Výhodou tohoto systému je, že se do sběru dat může zapojit prakticky kdokoli. Jedinou podmínkou umístění zařízení se zapnutou službou Bluetooth. U novějších zařízení je však důležité, povolit viditelnost zařízení tak, aby bylo viditelné i pro ostatní zařízení.



Obrázek 1: Funkce mýtné brány [2]

1.2 Využití dat

Získaná a zpracovaná data, mohou posloužit při vytváření modelů a simulaci dopravy. Lze tak poměrně přesně určit, v jaké denní době a na které části komunikace bude v budoucnosti docházet k dopravním komplikacím. Tato technika může být nápomocná v případě plánování dopravních uzavírek, trasy nadměrného nákladu a dalších událostí.

Dále lze z těchto dat vypočítat dojezdové doby do určitých lokací. Každý si tak může udělat přibližnou představu, jak dlouho bude trvat cesta z místa A do místa B v určenou denní dobu a den v týdnu. Nesmí se však zapomenout, že i přes sebelepší výpočet může

tuto predikci narušit nějaká nečekaná událost (dopravní nehoda, počasí, atd.) [5]. Systém na tomto principu je výborně vyřešen v Praze (viz kapitola 2.4.1).

Tyto data lze dále také použít při návrhu nových, případně zkapacitnění stávajících pozemních komunikací. Také mohou být využity pro plánování intenzity dopravy v řádech desetiletí. Pro tyto účely je však důležité počítat s koeficientem, který zohledňuje neustále přibývajícím množství vozidel [6]. Aby však výpočty a simulace byly k něčemu využitelné, je potřeba dlouhodobého sběru dat.

1.3 Šíření dopravních dat

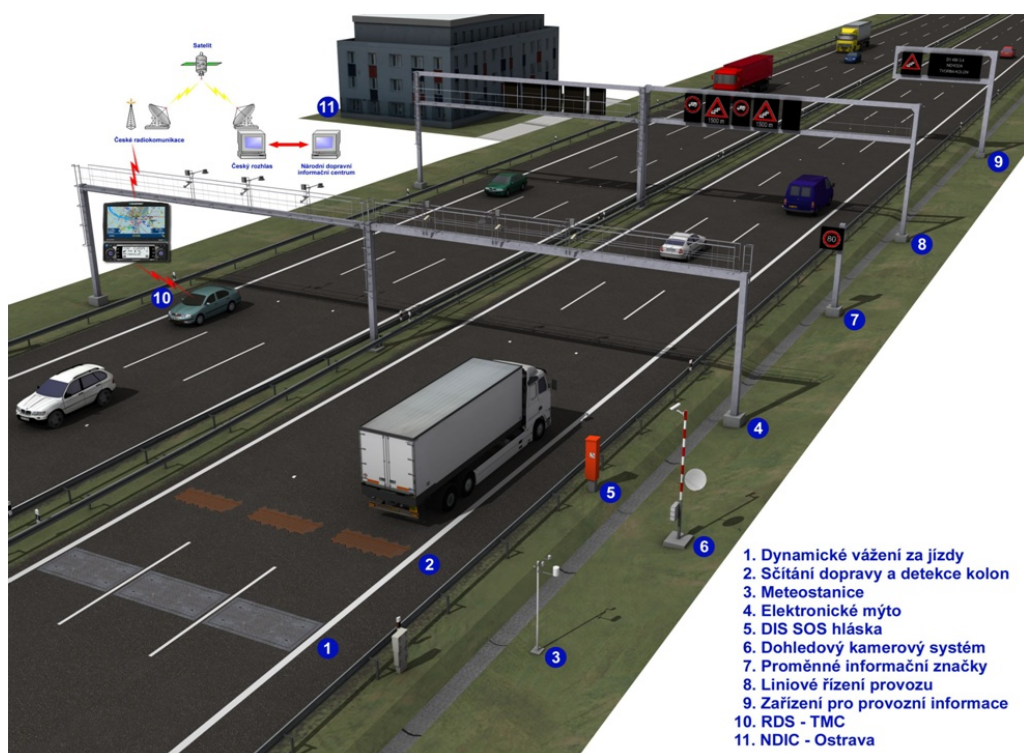
Dopravní data však nejsou pouze informace o intenzitě provozu. Existuje několik různých typů dopravních událostí, které na pozemních komunikacích mohou vznikat, ať plánovaně či nikoliv. Intenzita dopravy však může být na těchto událostech více či méně závislá.

Všechna tato data, ať už se jedná o výskyt neočekávané událostí (nehoda, nebezpečná situace apod.) tak o plynulosti dopravy by měla obyčejným lidem zajistit plynulejší a bezpečnější dopravu na pozemních komunikacích. Pokud má však tento princip nějakým způsobem fungovat, je důležité tato data lidem vhodným způsobem reprezentovat. K tomuto účelu existuje několik možností.

- **Proměnné dopravní značky** – tyto značky zobrazují informace o aktuální dopravní situaci přímo na pozemní komunikaci. Forma zobrazení je nejčastěji pomocí jednoho z osmy piktogramu dopravní značky. U některých značek je součástí i zařízení pro provozní informace. U tohoto zařízení lze přidat i doplňující informace na tři řádky, kde každý řádek může obsahovat až 15 znaků. Nevýhodou tohoto způsobu šíření informací je nízká četnost těchto cedulí, které jsou většinou nainstalovány jen na dálnicích a rychlostních silnicích. Naopak velkou výhodou je velmi rychlá aktualizace v případě výskytu dopravního problému. [7]
- **Liniové řízení provozu** – principem funkčnosti je velmi podobný předchozímu typu. Rozdílem však je, že tento systém je tvořen takzvanými portály, které jsou od sebe vzdáleny zhruba 1000 až 1500 metrů. Na každém z těchto portálů jsou umístěny proměnné dopravní značky, detektory měřící hustotu provozu, jeho intenzitu a průměrnou rychlost. Pomocí těchto informací zobrazuje na proměnné značce takové informace, aby jízda vozidel byla vždy co nejplynulejší a nejbezpečnější. Tyto portály se nejčastěji vyskytují v místech častých výskytů kolon a vysokých intenzit provozu. Dále jsou většinou vybudovány v okolí dálničních tunelů, kde mohou řídit vjezd vozidel dle jednotlivých pruhů. [8]
- **RDS-TMC** – doby, kdy GPS navigace byly součástí pouze dražších vozů, jsou již dávno minulostí. V současnosti existují desítky výrobců GPS navigací, které se tak staly dostupné prakticky pro kohokoliv. U navigací střední třídy lze většinou nalézt také funkci RDS-TMC. Pomocí této technologie lze přijímat aktuální dopravní informace. Tyto informace jsou přenášeny prostřednictvím neslyšitelného kanálu

RDS, a to formou zakódovaného toku dat TMC. Navigace po přijetí tyto data dekoduje a prakticky ihned po vzniku události umožní řidiči reagovat na nenadálou situaci na trase, nejčastěji nabídnutím objízdné trasy. [9]

- **Webové servery** – internet obsahuje velké množství informací a tak není až tak velkým překvapením, že na něm lze najít i informace o aktuální dopravní situaci. Výhodou této technologie šíření dat je její snadná dostupnost a množství obsažených detailů. Ještě před vyražením na cestu tak lze poměrně snadno naplánovat trasu s ohledem na aktuální dopravní situaci. Problém však nastává v případě výskytu neočekávané události. Na takovou událost již není možné žádným způsobem zareagovat.
- **Rozhlasové a televizní služby** – aktuální informace z pozemních komunikací jsou také často sdělovány pomocí rozhlasových a televizních stanic. Ty berou informace o dopravě jak z JSDI (viz kapitola 2.1), tak z aplikace ViaRodos (viz kapitola 2.2), ale i od dopravních a leteckých agentů, kteří události pomocí telefonu nahlásí. Někdy jsou tak informace o události mnohem výstižněji popsány, než strohá informace např. pomocí informační tabule.



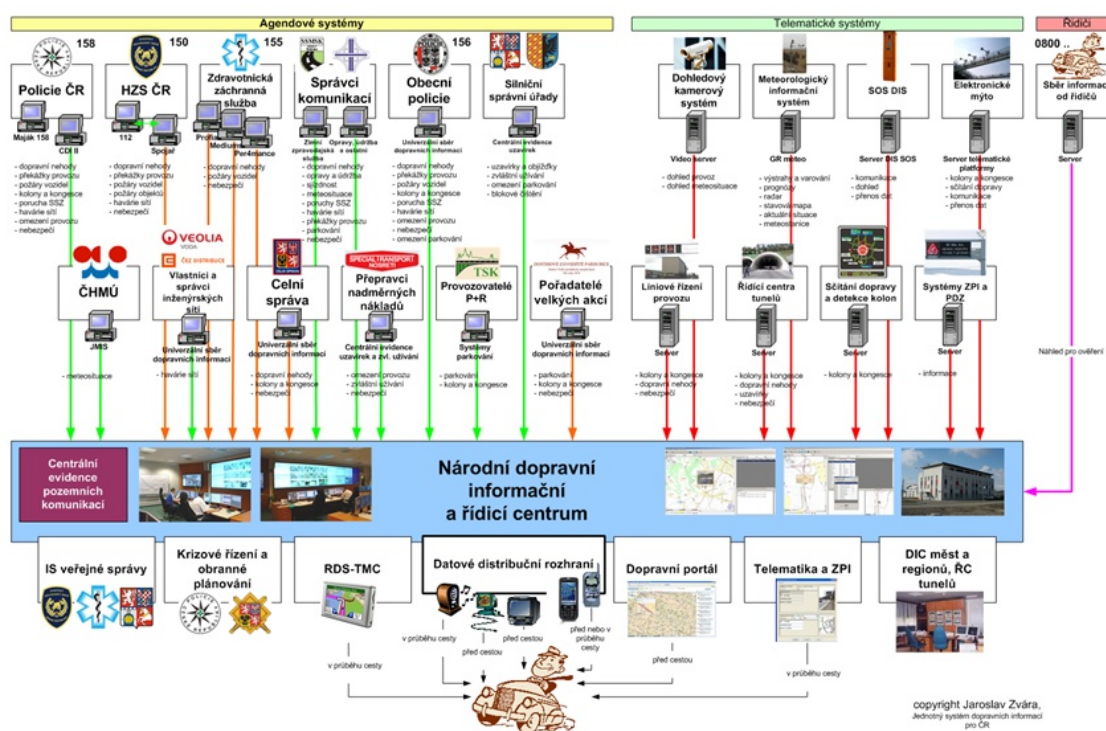
Obrázek 2: Zobrazení některých telematických systémů a systémů pro šíření dat [10]

2 Systémy dopravních dat

V České republice existují různé informační systémy, které poskytují data o aktuální dopravní situaci na pozemních komunikacích. V této kapitole jsou zmíněny nejdůležitější systémy a jejich principy, na kterých jsou založeny. Jsou však zmíněny i systémy, které data neposkytují, ale poskytují jistou unikátní informační hodnotu.

2.1 Jednotný systém dopravních informací

Hlavní a nejdůležitější zdroj dopravních dat v České republice poskytuje NDIC, což je technické, technologické a provozní pracoviště JSDI. Tento systém vznikl začátkem listopadu 2005, a to ve spolupráci Ministerstva dopravy, Ministerstva vnitra a Ředitelství silnic a dálnic. Hlavním úkolem tohoto systému je nepřetržitý sběr, zpracování, publikování a následná distribuce aktuálních dopravních informací. Princip systému je zobrazen na obrázku 3.



Obrázek 3: Schéma získávání dat JSDI [11]

Po zpracování NDIC zasílá tyto dopravní informace na proměnné informační tabule, webové stránky, rozhlasovým a televizním stanicím, telekomunikačním operátorům a

také přihlášeným odběratelům. Detailní dopravní informace jsou také neustále vysílány prostřednictvím služby RDS-TMC (viz kapitola 1.3). [11]

Zdroje dat

Jak je uvedeno na stránkách Ředitelství silnic a dálnic [12], NDIC využívá ke sběru dat následující zdroje:

- Policie ČR,
- Hasičský záchranný sbor,
- Zdravotnická záchranná služba,
- Správci komunikací všech kategorií,
- Silniční správní úřady všech úrovní,
- Obecní a městské policie,
- Český hydrometeorologický ústav.

Data jsou také získávána z různých čidel a telematických systémů. Tyto čidla dokážou zautomatizovat sběr důležitých dat. Mezi hlavní patří:

- Dohledový kamerový systém,
- Detekce intenzit dopravy,
- Silniční meteorologický systém,
- Systém elektronického mýta,
- Systém liniového řízení provozu SOKP,
- Systém sčítání dopravy, detekce kolon a sledování dopravního proudu,
- Řídicí systémy tunelů,
- Detekce jízdy vozidel v protisměru.[12]

Přístup k datům

Prvním předpokladem pro získání přístupu k aktuálním dopravním informacím je podepsání jednotné typové smlouvy. Uzavření smlouvy i následné zasílání dat je zdarma. To platí jak pro fyzické, tak i pro právnické osoby, které však musí zajistit další efektivní šíření těchto dat ve prospěch plynulosti a bezpečnosti silničního provozu. [13].

Data jsou z DDR zasílána pomocí několika dostupných protokolů. Na výběr je zasílání pomocí HTTP POST požadavku, na libovolný FTP server, případně na emailovou adresu pomocí protokolu SMTP. O technické části získávání dat pojednává kapitola 5.

2.1.1 Rozsah zasílaných dat

NDIC umožňuje na výběr zasílání dat v několika různých velikostech a nastaveních. Na výběr je k dispozici základní datová sada **basic**. Tato sada obsahuje pouze omezené informace o události, jako je textový popis, platnost události a lokalizace výskytu události.

Další a mnohem více datově obsáhlejší je rozšířená datová sada **extended**, která obsahuje všechny informace jako sada základní, ale přidává navíc i informace o typu události a ovlivněných úsecích pozemní komunikace (více v kapitole 3.2).

U obou datových sad lze zvolit formát, ve kterém bude NDIC zasílat informace o místě výskytu události. Lokalizace může být zasílána jako zeměpisné souřadnice v souřadnicových systémech WGS-84 nebo S-JTSK. Obě tyto metody jsou popsány v kapitole 3.3. U některých informací jsou zasílány i informace o výskytu dle registru UIR-ADR (viz kapitola 3.1.1).

Základní datová sada je tedy vhodnější k okamžité reprezentaci dat. Pro jednodušší systémy kde nebude potřeba filtrovat podle typu události nebo zobrazovat ovlivněné úseky je tato sada naprosto dostačující. Naopak u složitějších systémů je nutností použití datové sady **extended**.

Výběr dat podle typu

Dále je možné specifikovat, které typy dopravních událostí bude systém zasílat. Pokud nás tedy zajímají pouze informace o nehodách, systém nám bude zasílat pouze tento typ událostí. Typ přijímaných informací lze libovolně nakonfigurovat z následujících položek.

- **Dopravní nehody, dopravní události, kongesce** – vyskytují v náhodných lokalitách a v předem nepředvídatelných časech. Většinou trvají krátkou dobu v řádu jednotek minut, maximálně hodin.
- **Intenzita dopravy** – nejčastější typ dopravních informací přijímaných z DDR. Intenzita dopravy je poskytována hlavně na důležitých dopravních tepnách a ve velkých městech. Vyznačuje se velmi nízkým počtem nových událostí, ale velmi vysokou frekvencí aktualizací již dříve vytvořených událostí.
- **Uzavírky a omezení** – uzavírky lze rozdělit na plánované a neplánované. Většina uzavírek je plánovaných (oprava cesty, ulice, mostu atd.), ale mohou nastat i neplánované jako je například uzavření cesty z důvodu dopravní nehody.
- **Zimní informace** – v zimních měsících (přibližně od listopadu do února) NDIC poskytuje informace o sjízdnosti vozovek regionu jako celku. Seznam těchto regionů poskytuje jako samostatný číselník. Nemá tedy žádnou spojitost s UIR-ADR číselníkem popsaným v kapitole 3.1.1.

Výběr podle lokality

NDIC dále umožňuje konfiguraci území, pro které bude dopravní informace zasílat. To může být vhodné při vývoji systémů, které se budou zaměřovat pouze na určitou část republiky.

2.2 ViaRodos.cz

Zajímavou aplikací z pohledu zpracování dopravních dat je ViaRodos, kterou vytvořilo centrum RODOS. Toto centrum tvoří tři největší technické školy, jedna výzkumná instituce a šest dalších subjektů. I když na projektu spolupracuje několik subjektů z celé republiky, projekt je provozován pod záštitou Vysoké školy báňské - Technické univerzity Ostrava. Veškeré výpočty nad získanými daty jsou prováděny na ostravském superpočítaci, který je součástí centra excellence IT4Innovations. [14]

U tohoto systému je velmi zajímavá především technika, jakou dochází ke sběru dopravních dat. Základem pro sběr dat je přibližně 140 tisíc vozidel vybavených speciální jednotkou (viz GPS FCD v kapitole 1.1), která pomocí GPS zjišťuje data a v určitých časových intervalech je zasílá na server, který je zpracuje. Pro větší vzorek vozidel jsou však využívána i data z mýtných bran a dalších dopravních detektorů.



měrné rychlosti, zpoždění, apod. (viz obrázek 4). Lze tak jednoduše získat představu, jaké zpoždění nás na daném úseku čeká.

- **Města** – v systému jsou nyní zařazeny pouze tři největší města naší republiky. Po výběru města je zobrazen náhled s přehledným zobrazením dopravní intenzity. V náhledu jsou zobrazeny pouze důležité dopravní tahy a ulice.

2.3 CzechTraffic.cz

Systém pro sběr, analýzu a následné vyhodnocování dopravních dat. Dopravní informace jsou získávány jak z České republiky, tak i ze zahraničí. Tento systém je vyvíjen od roku 2007 ve spolupráci společností Telefónica O2 Czech Republic, a.s., SECAR BOHEMIA, a.s. a KAKTUS Software, s.r.o.

Data jsou získávána z několika různých zdrojů (viz obrázek 5). Hlavním zdrojem dat jsou vozidla vybavena speciální jednotkou (viz GPS FCD v kapitole 1.1). Mezi další zdroje patří JSDI (viz kapitola 2.1), dopravní agenti, kamerové záznamy a mobilní telefony.

Po získání a zpracování dat, systém umožňuje přístup k těmto datům pomocí API. O tento přístup je však nutné předem požádat, není však uvedeno zdali je tento přístup nějak zpoplatněn. [15]



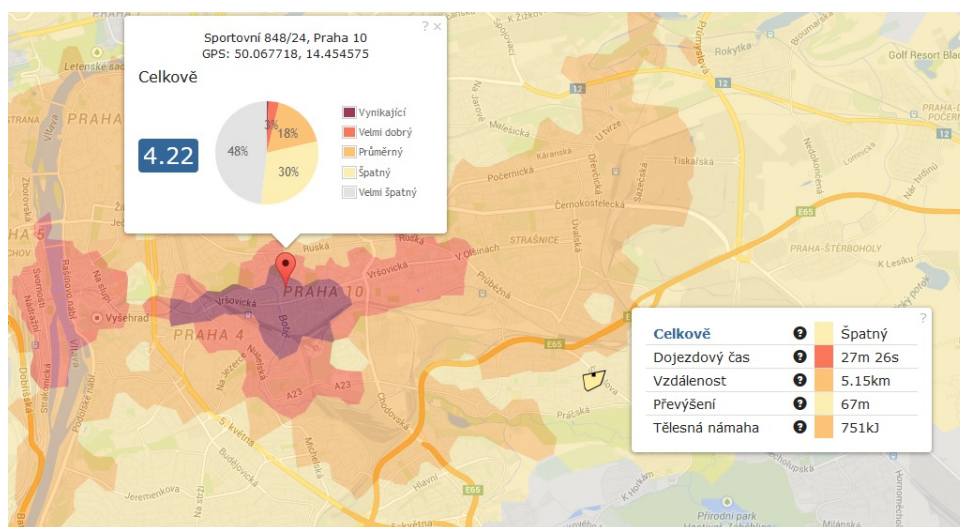
Obrázek 5: Zdroje dat a následné šíření u systému CzechTraffic.cz [15]

2.4 Lokální dopravní centra

Do této chvíle byly zmíněny pouze systémy pracující s daty v rámci celé České republiky. Existují však i systémy, které zpracovávají dopravní data pouze v určité části republiky. Tato část kapitoly je ve stručnosti věnována právě těmto systémům.

2.4.1 DopravniDostupnost.cz

Za vznikem tohoto projektu, stejně jako u aplikace ViaRodos (viz kapitola 4) stojí centrum RODOS. Hlavním cílem této aplikace je analyzování dostupnosti do jednotlivých částí města. Po kliknutí na libovolné místo ve městě, jsou v závislosti na vybraném dopravním prostředku, dnu v týdnu a čase dopočítány dojezdové doby do ostatních částí města. Ty jsou barevně zvýrazněny podle doby potřebné k dosažení tohoto místa. Po najetí myši nad vybrané místo je zobrazena tabulka znázorňující detailnější informace. Tento systém je nyní funkční pouze v Praze a italském Miláně.



Obrázek 6: Okno aplikace DopravniDostupnost.cz

Systém umožňuje přístup k těmto datům pomocí API, je však nutné o tento přístup požádat. Data tak mohou být vhodně využita například pro systémy dopravních společností, které po Praze, respektive Miláně rozvážejí zásilky. S pomocí těchto dat lze poměrně přesně naplánovat, jak rozvoz bude probíhat. [16]

2.4.2 Kamery.Praha.eu

Dalším zajímavou aplikací zobrazující dopravní informace je portál pražské dopravy. Jak již název napovídá, jedná se pouze o data ve městě Praha. U této aplikace zaujme především velké množství zobrazených informací. Od základních, které poskytují i jiné aplikace, jako je hustota dopravy, omezení a dopravní kamery až po pokročilé, které moc k vidění nejsou. Mezi ty se řadí například zobrazení pozic dopravních radarů, informace o parkovacích plochách a aktuální informace z několika meteo čidel. [17]

Bohužel i u tohoto systému jsem nenalezl možnost, jak se k těmto datům dostat.

3 Možnosti lokalizace události

Každá přijatá dopravní informace obsahuje určité informace, které určují její výskyt. Tato kapitola je věnována několika typům, dle kterých lze výskyt události určit.

3.1 Lokační tabulky

První možností, jak lze určit výskyt události je pomocí lokačních tabulek. Pomocí těchto tabulek lze jednoduše zjistit, v jaké administrativní jednotce se dopravní informace nachází. Tyto tabulky tedy neurčují konkrétní výskyt události, ale pouze informace o kraji, regionu, městu, ulici atd. Dále neobsahují žádné informace o typu ani úseku ovlivněné pozemní komunikace. Pro tyto účely jsou dostupné lokační tabulky pozemních komunikací (viz kapitola 3.2).

3.1.1 UIR-ADR

Tento registr byl vybudován v letech 1997-1999 za spolupráce obecních úřadů, Ministerstva pro místní rozvoj, Ministerstva vnitra, Českého úřadu zeměměřického a katastrálního, Českého statistického úřadu a České pošty s. p. Jedním z hlavních důvodů, proč tento registr vznikl, bylo sjednocení dat z několika menších registrů (ISEO-ADR, DDM, ISKN, RSO). Tato data je poté možné mnohem jednodušeji udržovat aktuální.

Ministerstvo práce a sociálních věcí ve spolupráci s obecními úřady spravuje registr adres u všech stavebních objektů, které mají určené domovní číslo. Aby registr zachoval určitou míru diskretnosti, neobsahuje žádné citlivé údaje o osobách ani organizacích. Ministerstvo ve spolupráci s obcemi průběžně doplňuje chybějící adresy, zaznamenává změny názvů, případně označuje zrušené stavební objekty.

Registr původně vznikl pouze pro potřeby informačních systémů ministerstva, po dokončení a otestování byl však tento registr uvolněn i pro veřejnost. Ministerstvo práce a sociálních věcí poskytuje tento registr veřejnosti k dispozici zdarma. První možností přístupu do registru je využití webových stránek ministerstva ¹. U tohoto přístupu není nutná žádná zvláštní softwarová výbava, ale prohlížeč musí splňovat určité parametry ². Další možností je zdarma získat CD-ROM s daty. Součástí CD-ROM je i aplikace umožňující vložení těchto dat do různých databázových systémů.

Od 1. července 2012 tento registr plně nahradil nový systém nazvaný RUIAN (viz kapitola 3.1.2). Rozdílové soubory jsou však i nadále vytvářeny přímo z registru RUIAN. V dohledné době se však počítá s úplným ukončením poskytování služeb registru UIR-ADR. [18]

3.1.2 RUIAN

Datový registr UIR-ADR postupem času nahradil registr RUIAN. Tento nový systém obsahuje stejná data jako jeho předchůdce, navíc byl ale doplněn o katastrální území, par-

¹<http://forms.mpsv.cz/uir/prohlizec/first.html>

²<http://forms.mpsv.cz/uir/prohlizec/second.html>

cely aj. Dalším významným milníkem je přidání grafických prvků, jako jsou definiční body, linie ulic a hranic. Stejně jako jeho předchůdce vznikl sloučením několika menších registrů (ISKN, RSO, DDM, REKOS, ZABAGED, ISEO-ADR, UIR-ADR). [19]

Data z tohoto registru jsou zdarma přístupné veřejnosti pomocí veřejného dálkového přístupu. Registr však umožňuje i stažení zdrojových dat ³ ve výměnném formátu VFR.

3.2 Lokační tabulky pozemních komunikací

Do této chvíle byly zmíněny pouze lokační tabulky na úrovni obcí, regionů a krajů. I tyto registry sice poskytují informace o výskytu události, nicméně neposkytují žádné informace o pozemních komunikacích. Nelze tedy určit přesný úsek pozemní komunikace, na který se dopravní událost vztahuje. NDIC však u většiny informací zasílá informace o ovlivněných úsecích. Z těchto dat lze poté přesně určit, který úsek komunikace a v jaké délce je ovlivněn. Při znalosti těchto informací je daleko jednodušší naplánovat objízdnou trasu, případně zobrazit ovlivněný úsek na mapě. Právě pro tyto účely jsou dostupné lokalizační tabulky pozemních komunikací.

3.2.1 Global Network

Jedná se o jednotnou georeferenční síť pozemních komunikací. Výrobce této geodatabáze je společnost CEDA ve spolupráci s ředitelstvím silnic a dálnic. Data pro tuto geodatabázi jsou získávána z centrální evidence pozemních komunikací [20]. Aktualizace těchto dat probíhá pravidelně dvakrát do roka. U ulic a veřejných prostranství je garantováno minimálně 99.4% pokrytí. Toto porovnání je prováděno vůči referenčnímu systému RUIAN (viz kapitola 3.1.2).

Obsah geodatabáze

Tato geodatabáze obsahuje následující typy pozemních komunikací:

- dálnice,
- silnice,
- ulice,
- místní a účelové komunikace,
- polní a lesní cesty.

Každý typ komunikace obsahuje popisující parametry. Global Network lze tedy použít pro naplánování trasy pro různé dopravní prostředky.

Global Network dále obsahuje i informace o rozestavěných úsecích, které budou uvedeny do provozu před vydáním následující verze. V současné době systém obsahuje přes 320 000 km pozemních komunikací [21].

³<http://vdp.cuzk.cz/vdp/ruian/vymennyformat/vyhledej>

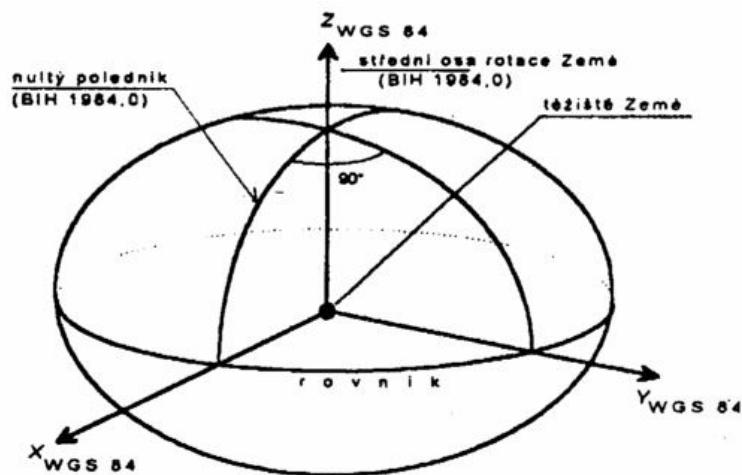
3.3 Souřadnicové systémy

Další možností, kterou lze určit výskyt události, je pomocí zeměpisných souřadnic. Souřadnice jsou někdy také nazývány jako koordináty. Dříve, než je však možné určit polohu objektu na Zemi, je potřeba znát, který druh soustavy souřadnic byl použit. Existuje totiž několik různých souřadnicových systémů, pomocí kterých lze označit stejné místo na zemském povrchu. Výsledné souřadnice však budou odlišné. K tomuto označení polohy je potřeba dvou souřadnic [22]. V České republice se nejčastěji lze setkat se souřadnicovými systémy WGS-84 a S-JSTK.

3.3.1 WGS-84

Tento souřadnicový systém je světově uznávaným geodetickým standardem. Referenční plocha systému je elipsoid WGS 84 s počátkem v hmotném středu Země. Jako kartografické zobrazení se využívá UTM. Osy X a Y leží v rovině rovníku, osa Z je stejná jako osa rotace Země v roce 1984. Počátek a orientace jeho os X, Y, Z jsou realizovány pomocí 12 pozemských stanic s přesně známými souřadnicemi. Tyto stanice nepřetržitě monitorují dráhy družic systému GPS. [23]

Souřadnicový systém WGS-84 je využíván při určování polohy u systému GPS, dále je hojně využíván u všech armád států, které jsou součástí NATO. Také byl použit jako primární souřadnicový systém v celé této práci.



Obrázek 7: Schéma WGS-84 zobrazení (zdroj: <http://gis.zcu.cz>)

GPS

Globální polohovací systém je navigační systém, pomocí kterého lze kdekoliv na světě určit svou přesnou polohu. Určování polohy funguje za jakéhokoliv počasí a není závislé ani na denní době. Výsledná poloha je určována v souřadnicích zeměpisné šířky a délky.

Původně se s tímto systémem počítalo pouze pro vojenské účely. Jak se však systém dále rozvíjel, začal být od 90. let dostupný po celém světě. Kongres spojených států tak umožnil používání GPS i v civilní sféře. S tímto povolením však vznikla také nové omezení, které znemožňovalo tuto technologii zneužít například pro teroristické útoky. Tímto omezením bylo záměrné zhoršování přesnosti. Uživatel tedy mohl nechat zaměřit svou polohu, ale s poměrně velkou odchylkou od skutečnosti. V současnosti je již záměrné zhoršování polohy vypnuto. Pro všechny civilní obyvatelé se tak několikanásobně zvedla přesnost lokalizace. Celá tento systém je rozdělen na tři části.

- **Kosmická část** – obsahuje 24 družic, z toho tři jsou záložní. Družice krouží ve výšce přibližně 20 000 km na 6 oběžných drahách. Každá družice obsahuje vysílač, přijímač, atomové hodiny a řadu dalších přístrojů. Každá družice dále zpracovává informace zasílané z pozemního řídicího centra. Na základě těchto informací může korigovat svou dráhu pomocí raketových motorů. Do řídicího centra dále každá družice zasílá stav vlastních systémů. V případě problémů, je každá družice vybavena záložními zdroji a bateriemi, které jsou dobíjeny dvěma slunečními panely.
- **Řídicí část** – tvoří hlavní řídicí stanice v Colorado Springs. Dále obsahuje další tři řídicí stanice, které spolupracují s hlavní stanicí. Náplní práce u této části je monitorování funkcí každé družice, sledování a výpočet dráhy letu, údržbové práce, především údržba přesného času atomových hodin.
- **Uživatelská část** – tu tvoří již samotní uživatelé s GPS přístroji. Uživatelé si často mylně myslí, že jejich zařízení aktivně komunikuje s družicemi. Ve skutečnosti je to přesně naopak. Družice zasílají pouze signál k Zemi, přístroj tyto data přijme a zpracuje. Přístroj je tedy zcela pasivní.

Určování polohy

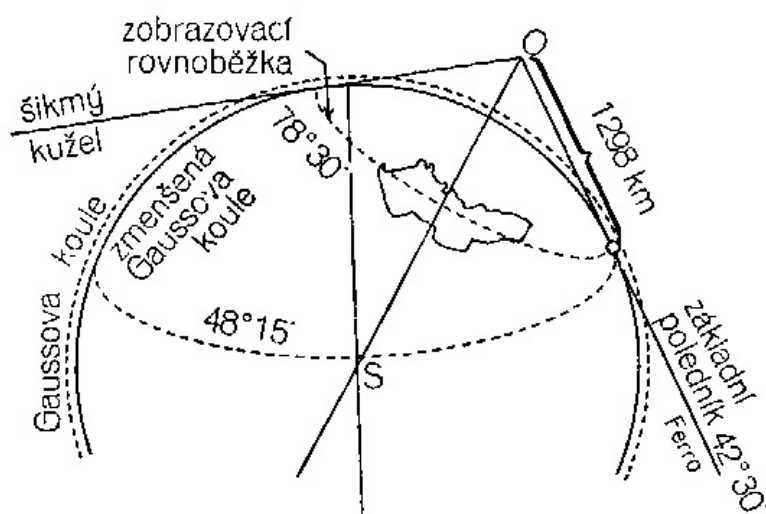
Princip určování polohy pomocí GPS je v principu snadno pochopitelný. Družice vysílají směrem k Zemi signály pro uživatele v podobě složitějšího signálu. Tyto signály obsahují informace o poloze družice, která signál vysílala a přibližné pozice ostatních družic. K určení aktuální polohy přijímací zařízení vypočítá tzv. pseudo vzdálenosti. Tyto pseudo vzdálenosti jsou vzdálenosti mezi přijímačem a viditelnými družicemi. Tento výpočet je založen na znalosti rychlosti šíření družicového signálu a rozdílu času mezi vysláním a příjmem signálu. Pro určení dvojrozměrné polohy stačí příjem signálu ze tří družic. Pro určení nadmořské výšky je potřeba alespoň čtyř družic. [24]

3.3.2 S-JSTK

Systém jednotné trigonometrické sítě katastrální je pravoúhlý souřadnicový systém využívaný na území České republiky a Slovenska. Základem pro tento souřadnicový systém je jednotná trigonometrická síť katastrální, což je síť geodetických bodů budovaných v letech 1920 až 1957.

Systém vychází z tzv. Křovákova zobrazení, které definoval Ing. Josef Křovák v roce 1922. Ten dostal za úkol navrhnout takové zobrazení, které by nejvíce vyhovovalo tehdejší Československé republice. Jako počátek pravoúhlé rovinné soustavy je zvolen obraz vrcholu kužele. Osa X je tvořena obrazem základního poledníku a její kladný směr je orientován k jihu. Osa Y je kolmá k ose X a směřuje na západ. Tímto zobrazením se tak obě republiky dostali do prvního kvadrantu a obě souřadnice jsou tedy kladné. Dále platí, že pro libovolný bod na území bývalého Československa je Y souřadnice vždy menší než souřadnice X. [25]

Souřadnicový systém S-JTSK je používán jako závazný pro všechna státní mapová díla.



Obrázek 8: Schéma Křovákova zobrazení [25]

4 Návrh modelu

Tato kapitola je zaměřena na návrh databázových modelů. V první podkapitole je ukázán návrh řešení modelu na obecné úrovni. Další část kapitoly je věnována návrhu modelu, který byl použit při vlastní implementaci.

4.1 Obecný návrh modelu

Součástí zadání této práce je i požadavek na návrh obecného modelu, který by umožnil agregování dopravních informací z několika různých zdrojů dat. Tento obecný model by měl zvládat několik typů dopravních informací, a tyto informace by měli být vhodně ukládány pro další práci nad těmito daty. Motivací pro vytváření tohoto modelu je především velké množství dopravních informací, nad kterými by poté bylo možné různě experimentovat.

4.1.1 Typy pozemních komunikací

Dříve než bude možné specifikovat typy dopravních informací, je důležité zjistit, jaké typy pozemních komunikací v České republice existují. Tabulka 1 obsahuje seznam všech typů seřazených podle priority od nejvyšší. U každého typu komunikace je dále uveden vlastník a objekt zodpovědný za údržbu.

Typ	Vlastník	Správa a údržba
Dálnice	stát	ŘSD ČR
Rychlostní silnice	stát	ŘSD ČR
Silnice 1. třídy	stát	krajské SÚS, smluvní firmy
Silnice 2. třídy	kraj	krajské SÚS, smluvní firmy
Silnice 3. třídy	kraj	krajské SÚS, smluvní firmy
Místní komunikace	města a obce	technické služby, smluvní firmy
Veřejně přístupné účelové komunikace	privátní, veřejné osoby	smluvní firmy

Tabulka 1: Typy pozemních komunikací [26]

4.1.2 Typy informací

V dalším kroku je důležité zjistit, jaké typy dopravních informací mohou na pozemních komunikacích vznikat, a jak bude definován jejich výskyt. To souvisí i s výše uvedenou tabulkou 1, jelikož některé typy informací jsou vázány pouze na určité typy komunikací.

Dopravní události, kongesce

Tento typ události je nejčastějším, který na pozemních komunikacích vzniká. Obsahuje všechny typy událostí od dopravních nehod, omezení, nebezpečných situací až po práce na silnici. Vyskytují se na všech třídách pozemních komunikací (viz tabulka 1). Většina událostí obsahuje i informace o administrativní jednotce a ovlivněných úsecích pozemní komunikace. Události lze dále rozlišit podle toho, zdali je výskyt události plánován či nikoliv.

- **Plánované** – u plánovaných událostí je předem znám termín od kdy do kdy bude tato událost platná. Většinou se jedná o různé práce na silnici, omezení a zákazy atp. Platnost tohoto události tohoto typu je většinou v rámci dnů až týdnů.
- **Neplánované** – neplánované jsou takové události, u kterých nelze určit přesný počátek a konec platnosti události. Vyskytují se v náhodných lokacích. Délka trvání události je většinou v rámci hodin, maximálně dnů.

Plynulost dopravy

Tento typ události je znám spíše pod pojmem stupně dopravy. Obsahuje informace o plynulosti dopravy na pozemních komunikacích. Tyto informace jsou zasílány především z dálnic, rychlostních silnic a některých silnic 1. třídy (viz tabulka 1). Vyznačuje se častou aktualizací dat. Tyto data obsahují informace o úsecích pozemní komunikace, které jsou většinou neměnné. Technologie sběru těchto dat jsou popsány v kapitole 1.1. Plynulost dopravy lze rozdělit do následujících pěti stupňů.

- **Stupeň 1** – pohyb pouze jednotlivých vozidel, průměrná rychlost není oproti maximální povolené nijak snížena. Jízda je plynulá.
- **Stupeň 2** – pohyb malých skupin vozidel, nedochází ke vzniku kolon, odbavování na světelných křižovatkách je plynulé. Jízda je plynulá.
- **Stupeň 3** – pohyb proudů vozidel, odbavování na světelných křižovatkách již není plynulé. Jízda je plynulá, ale v žádném úseku již nedosahuje maximální povolené rychlosti.
- **Stupeň 4** – po pozemní komunikaci se pohybují kolony vozidel, odbavování na světelných křižovatkách je výrazně narušeno a není plynulé. Jízda není plynulá a vyznačuje se velmi nízkou průměrnou rychlostí.
- **Stupeň 5** – stání, případně velmi pomalá jízda kolony vozidel. Na světelných křižovatkách dochází k vytváření dlouhých kolon. Jízda není plynulá a průměrná rychlost klesá na minimum. Lze označit jako dopravní kolaps. [27]

Počasí

Nedílnou součástí, která může ovlivňovat výskyt dopravních událostí, jsou informace o počasí. Při návrhu modelu je počítáno s ukládáním informací o počasí, pro jednotlivé

okresy naší republiky. Tato velikost oblastí by měla být pro většinu okresů naprosto dostačující, jelikož na takto velkém území většinou nedochází k významným rozdílům.

Problém však může nastat v zimních měsících u okresů, které obsahují níže položené oblasti a zároveň i vysoko položené hory. Pro tyto situace model umožňuje vložení volitelné oblasti, pro kterou je zadané počasí platné. V rámci jednoho okresu tak lze vytvořit několik vlastních oblastí. V případě, že by informace o počasí byly zjišťovány pomocí meteo stanice, pozice této stanice by byla definována zemskými souřadnicemi.

Sjízdnost vozovek

Sjízdnost vozovek navazuje na předchozí typ, tedy informace o počasí. Rozdílem však je, že informace o počasí mohou být dostupné v průběhu celého roku. Sjízdnost vozovek je důležitá pouze v měsících, kdy hrozí zhoršená sjízdnost vozovek, tedy především v zimních měsících.

Dalším rozdílem je způsob lokace události. Zatímco informace o počasí je vázána na konkrétní region, případně oblast, u sjízdnosti je informace vázána na konkrétní úsek komunikace dle lokačních tabulek pozemních komunikací (viz kapitola 3.2). U každého záznamu jsou ukládány informace určující sjízdnost vozovky, dále jsou uloženy informace poskytující přehled o povrchu vozovky. Volitelnými informacemi každé události jsou informace o počasí (vítr, teplota, apod.).

Snímky z kamer

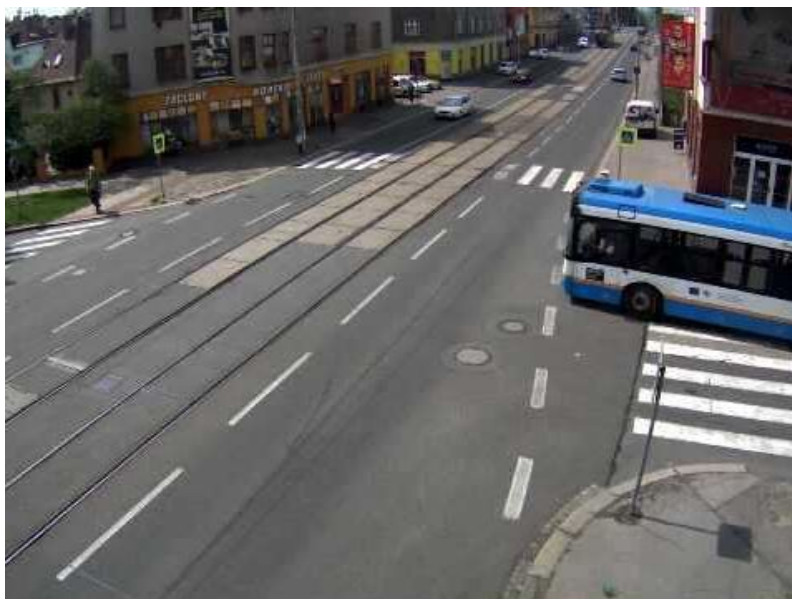
Tento typ informace má spíše informativní charakter. Umožňuje pouhým pohledem určit stav dopravy na pozemních komunikacích. V modelu je tento typ vložen jako kterákoliv dopravní událost s tím rozdílem, že umožňuje vložení datových souborů obsahujících snímky z kamery. Každá verze tohoto typu události může obsahovat několik kamerových snímků. Pozice kamery je uložena jako zeměpisné souřadnice (viz kapitola 3.3.1).

Každý vložený snímek však zabírá několikanásobně více místa na pevném disku než kterýkoliv jiný typ dopravní informace. Postupem času by tak došlo k velkému nárůstu velikosti celé databáze. To by vedlo k neustálé nutnosti přidávání datových kapacit. Řešením jak se tomuto problému vyhnout, je ponechání kamerových snímků pouze pro verze vytvořené v posledních 24 hodinách. U starších by docházelo k automatickému odebrání nahraného snímku. Náhled kamerového snímku je zobrazen na obrázku 9.

Parkovací místa

Pomocí tohoto typu události lze na mapě zobrazit seznam parkovišť společně s informací o aktuálním počtu volných parkovacích míst. U parkovišť, vybavených automatickým odbavováním by každé vozidlo vytvořilo u daného parkoviště záznam, který by pouze určoval čas odbavení a směr (vjezd, výjezd). Pomocí těchto záznamů by poté bylo jednoduché určit počet volných míst.

Při vyšším počtu parkovišť zapojených do tohoto systému, by však docházelo ke generování velkého množství záznamů. Alternativou pro tento systém, která nakonec byla využita při návrhu, je zasílání informací o parkovacích místech v určitém časovém intervalu. V každém tomto intervalu by byla zaslána informace, kolik vozidel vjelo do

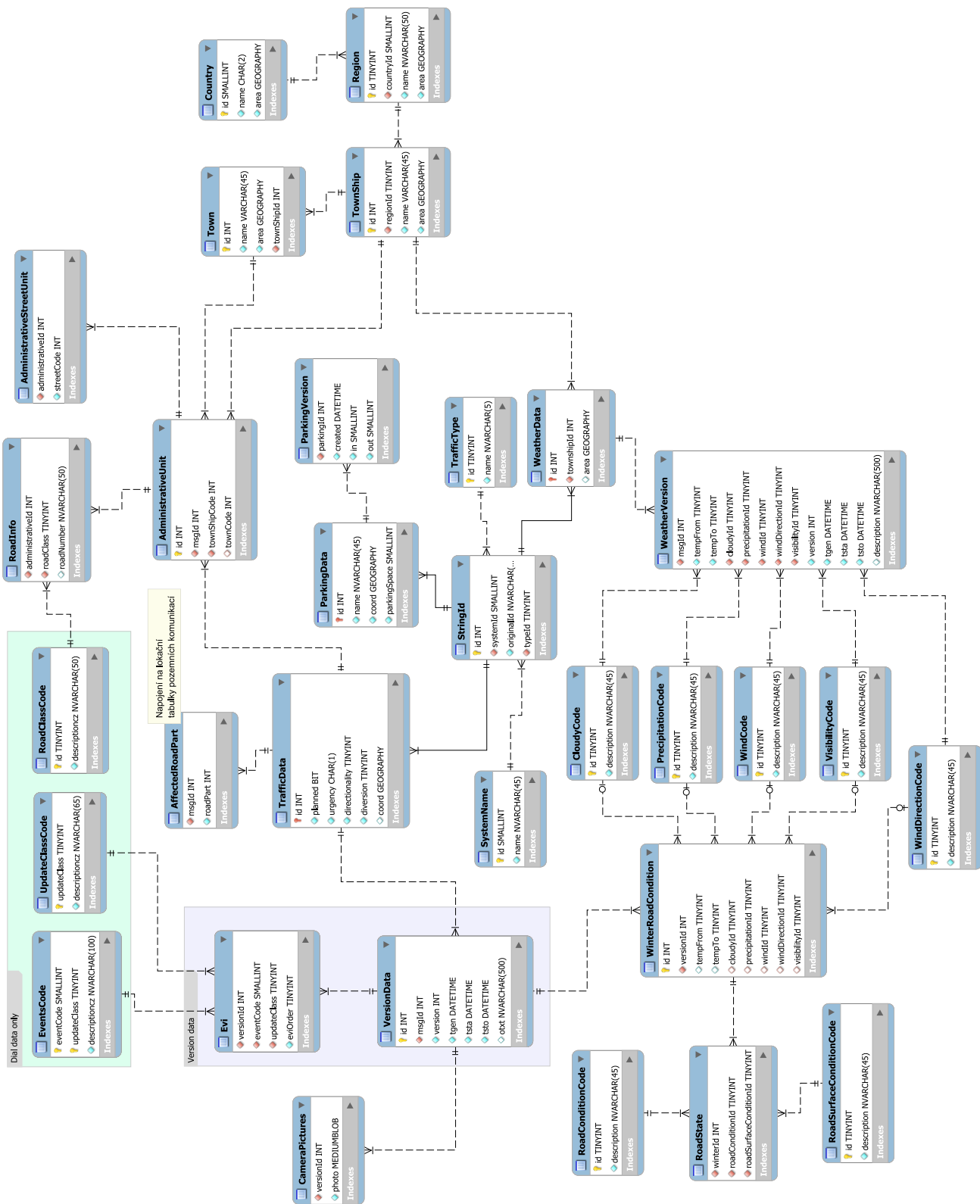


Obrázek 9: Kamerový snímek z Ostravy (zdroj: <http://mapy.ovanet.cz/krizovatky>)

parkoviště, a kolik jich naopak parkoviště opustilo. Tento systém by stejně jako předchozí návrh umožnil monitorovat počet volných parkovacích míst, ale vytvářel by nezanedbatelně nižší počet záznamů. Pozice parkoviště na mapě je definována zeměpisnými souřadnicemi (viz kapitola 3.3.1).

4.1.3 ER diagram

ER diagram obecného modelu je zobrazen na obrázku 10. Pomocí tohoto typu diagramu lze přehledně zobrazit jednotlivé entity a jejich vztahy mezi sebou. Současně je dostupný jako obrázek ve vysokém rozlišení na přiloženém DVD.



Obrázek 10: ER diagram obecného modelu

4.1.4 Vlastnosti modelu

Model je navržen tak, aby v nezávislosti na systému umožnil vložení dopravních informací. Problém by však mohl nastat v případě, kdy systém bude vyplňovat informace o události, jako typ události, administrativní jednotku, ovlivněné úseky komunikace atp. Pro tyto případy by bylo důležité dodržování standardních číselníků.

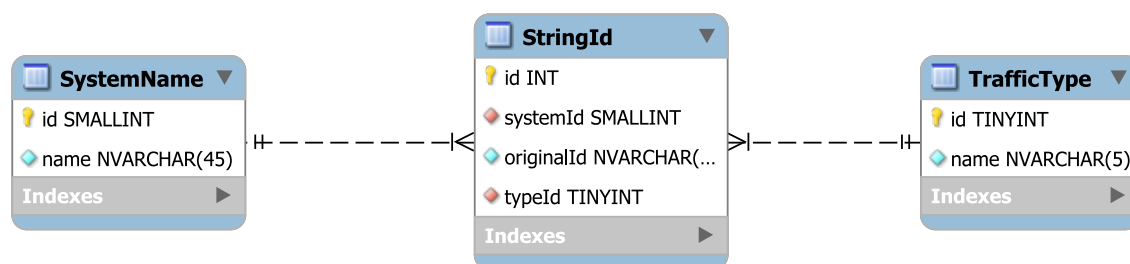
Pro zápis informací o typu události je v Evropě standardem číselník Alert-C. Tento číselník využívá při vytváření událostí i NDIC (viz kapitola 2.1). Také je využíván při zasílání informací skrze RDS-TMC. [28]

Informace o administrativní jednotce, které událost náleží, by měla být uložena pomocí lokačních tabulek UIR-ADR nebo RUIAN (viz kapitola 3.1). Ovlivněné úseky pozemních komunikací podle lokačních tabulek pozemních komunikací (viz kapitola 3.2). U dalších číselníků by bylo nutné dodržovat číslování, které by bylo definováno v dokumentaci.

4.1.5 Vkládání dat

Obecný datový model počítá se vkládáním událostí z více systémů. Každý ze systémů navíc může zasílat různé typy událostí s různým formátem jednoznačného identifikátoru. Pokud by však nebylo žádným způsobem rozlišeno, z jakého systému událost dorazila, mohlo by docházet k porušení konzistence dat.

Prvním krokem je tedy registrace systému do tabulky `SystemName`. Při vkládání události by tímto bylo zaručeno, že si systémy nebudou navzájem přepisovat data. Může však nastat i situace, že v rámci jednoho systému bude pro různé typy událostí generován stejný identifikátor. Je tedy potřeba brát v potaz i tuto možnost. Výsledný identifikátor události (`id` z entity `StringId`), který je využíván skrze celý datový model by se tedy vytvářel z ID systému (`systemId`), ID typu události (`typeId`) a původního přijatého identifikátoru. To vše by bylo uloženo v tabulce `StringId` (viz obrázek 11) Nad těmito daty je dále vytvořen složený unikátní klíč.

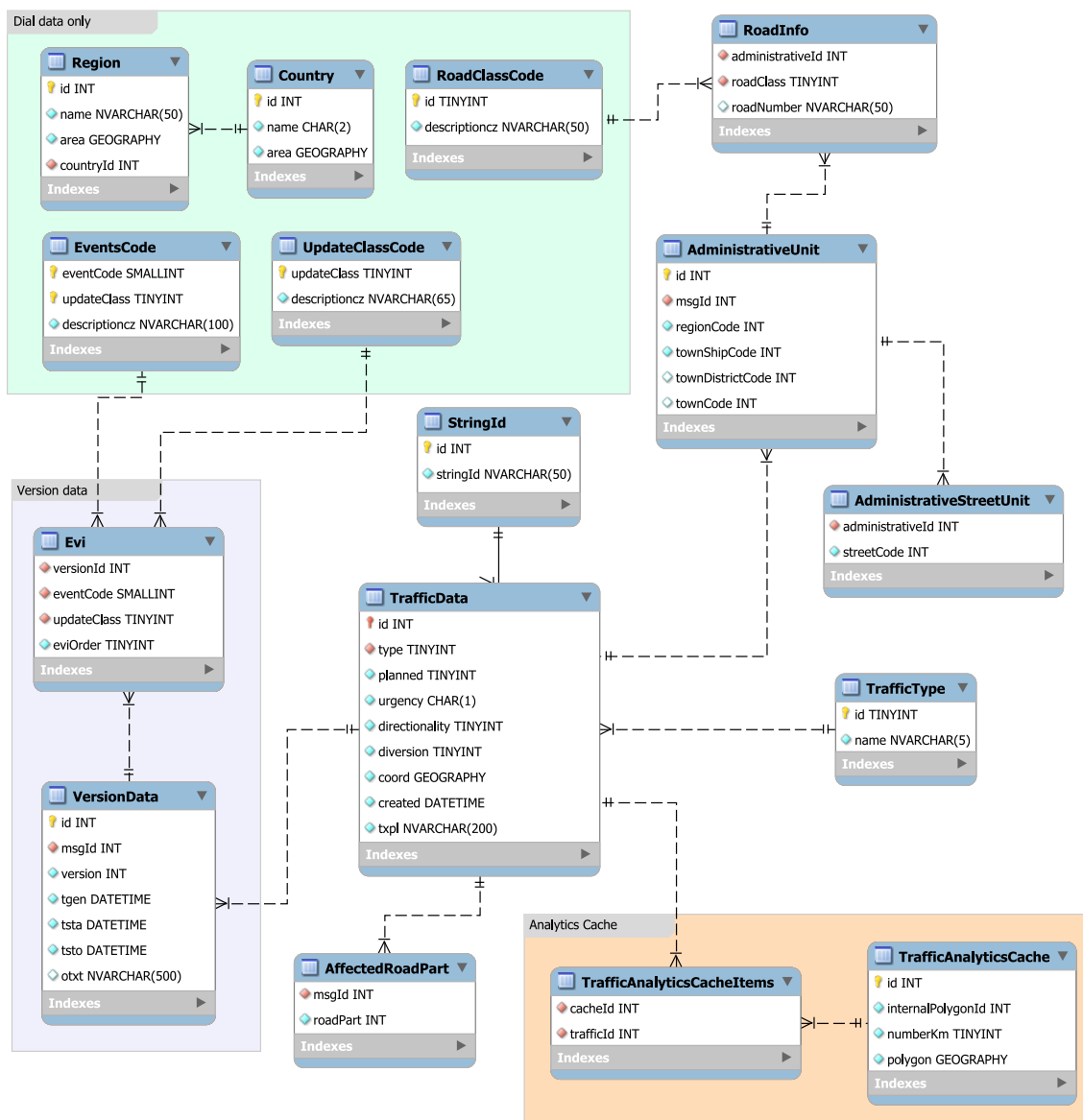


Obrázek 11: Vkládání dat z různých systémů

4.2 Realizovaný datový model

V předchozí části byl ukázán návrhu obecného databázového modelu, který by umožnil ukládání dat z více systémů. Tato část kapitoly je již věnována návrhu databázového modelu, který je využit při praktické implementaci.

4.2.1 ER Diagram



Obrázek 12: ER Diagram realizovaného modelu

4.2.2 Datový slovník

V datovém slovníku jsou zobrazeny jednotlivé databázové entity použité při implementaci. Každá tabulka obsahuje samostatnou entitu, její atributy datové typy, a další důležité informace. Pro přehlednost byl datový slovník zařazen do přílohy B.1.

4.2.3 Lineární schéma entit

Název	(primární klíč, cizí klíč, atributy)
AdministrativeStreetUnit	(<i>administrativeId</i> , <i>streetCode</i>)
AdministrativeUnit	(<i>id</i> , <i>msgId</i> , <i>regionCode</i> , <i>townShipCode</i> , <i>townDistrictCode</i> , <i>townCode</i>)
AffectedRoadPart	(<i>msgId</i> , <i>roadPart</i>)
Country	(<i>id</i> , <i>name</i> , <i>area</i>)
EventsCode	(<i>eventCode</i> , <i>updateClass</i> , <i>descriptioncz</i>)
Evi	(<i>versionId</i> , <i>eventCode</i> , <i>updateClass</i> , <i>eviOrder</i>)
Region	(<i>id</i> , <i>name</i> , <i>area</i>)
RoadInfo	(<i>administrativeId</i> , <i>roadClass</i> , <i>roadNumber</i>)
RoadClassCode	(<i>id</i> , <i>descriptioncz</i>)
StringId	(<i>id</i> , <i>stringId</i>)
TrafficAnalyticsCache	(<i>id</i> , <i>internalPolygonId</i> , <i>numberKm</i> , <i>polygon</i>)
TrafficAnalyticsCacheItems	(<i>cacheId</i> , <i>trafficId</i>)
TrafficData	(<i>id</i> , <i>type</i> , <i>planned</i> , <i>urgency</i> , <i>directionality</i> , <i>diversion</i> , <i>coord</i> , <i>created</i> , <i>txpl</i>)
TrafficType	(<i>id</i> , <i>name</i>)
UpdateClassCode	(<i>updateClass</i> , <i>descriptioncz</i>)
VersionData	(<i>id</i> , <i>msgId</i> , <i>version</i> , <i>tgen</i> , <i>tsta</i> , <i>tsto</i> , <i>otxt</i>)

4.2.4 Lineární schéma závislostí

Součástí návrhu je i lineární schéma závislostí (viz tabulka B.2). Pomocí tohoto schématu lze snadno popsat vztahy mezi jednotlivými entitami.

4.2.5 Rozdíly mezi obecným a realizovaným modelem

Na první pohled je patrné, že realizovaný datový model je zjednodušenou verzí modelu obecného. V realizovaném modelu nejsou uloženy žádné informace o počasí, sjízdnosti vozovek, parkovacích místech ani snímky z kamer. Implementovaný model tak vznikl v závislosti na rozsahu dopravních informací, která jsou přijímána z JSDI (viz kapitola 5).

5 Získávání dat z JSDI

Jak bylo uvedeno v kapitole 2.1 v České republice existuje jednotný systém dopravních informací. Tato kapitola je věnována technickým aspektům získávání dat právě z tohoto systému.

5.1 Příjem dat

NDIC informace zasílá uložené ve standardním datovém formátu XML. Tento XML má definovanou určitou strukturu, která je popsána v dokumentaci ⁴. Počet událostí obsažených v XML souboru není pevně určen a je závislý především na množství událostí na silnicích. NDIC také umožňuje nastavit, maximální velikost souboru, který bude vytvářen.

Před uvedením ostrého provozu distribuce dopravních informací pomocí DDR je provedeno otestování komunikace mezi poskytovatelem a odběratelem [29].

Systém informace zasílá skrze DDR, což je rozhraní starající se o správnou distribuci dopravních informací. Předtím, než bude možné data přijímat, je důležité vybrat technologii pro příjem dat. Data mohou být přijímána pomocí následujících technologií.

HTTP

Jak už název služby napovídá, jedná se o zasílání dat pomocí webového protokolu HTTP. Systému je nahlášena URL adresa serveru a skriptu. Na tuto adresu jsou poté zasílány HTTP POST požadavky, kde součástí tohoto požadavku jsou požadovaná data. NDIC preferuje právě tento protokol pro zasílání zpráv. Lze však zvolit i některou z dalších technologií.

FTP

Tento systém příjmu dat je založen na protokolu FTP. Tento protokol slouží pro ukládání souborů na server. Systému je nahlášena FTP adresa serveru, na který budou data nahrávána. Pokud jsou pro uložení dat nutné i autorizační informace (uživatelské jméno a heslo), musí být zaslány i tyto údaje. Po obdržení všech těchto údajů ukládá systém XML soubory s daty do námi specifikované složky.

Tento způsob příjmu dopravních informací byl zvolen při vlastní implementaci, která je popsána v kapitole 6.2

SMTP

Poslední metodou, kterou NDIC umožňuje vybrat, je příjem informací pomocí protokolu SMTP. Systému je nahlášena emailová adresa, na kterou má informace zasílat. Dopravní informace jsou přiloženy k emailu jako příloha a následně na tuto adresu zaslána. Tento způsob přenosu informace může být v určitých případech docela nespolehlivý a problematický (spam filtry, blacklist, náhodné pořadí). Tento způsob přenosu bych tedy důrazně nedoporučoval. [30]

⁴http://www.dopravniinfo.cz/public/data/file/Rozhrani_DDR_v3.2.5.pdf

5.2 Validace dat

Před samotným zpracováním přijatých dat, je vhodné provést kontrolu. Tato kontrola zaručí, že přijatá data jsou validním XML dokumentem a obsahují všechny informace, co obsahovat mají. Bohužel NDIC odběratelům k dispozici neposkytuje žádné W3C XML schéma, dle kterého by bylo možno validaci provádět. Jediným řešením jak toto schéma opatřit, je jeho vlastní výroba podle dostupné dokumentace. Tento způsob byl použit i při implementaci. Schéma je dostupné na přiloženém DVD, je však důležité upozornit, že toto schéma bylo vytvářeno na míru pouze pro tuto implementaci, nepočítá tak například s příjmem informací o sjízdnosti vozovky.

5.3 Výpadky komunikace

U každé komunikace se může stát, že server nebo klient nejsou z nějakého důvodu dostupní. DDR myslí i na tyto případy a umožňuje zaslání informací i v případě tohoto výpadku.

Pokud je zvolena služba na protokolech FTP nebo HTTP, systém se bude pokoušet data opakovaně odesílat až do doby, kdy dojde k opětovnému navázání spojení.

U emailového odběru pomocí protokolu SMTP se systém opakovaně snaží email s daty odeslat.

Dopravní informace, které se nepodařilo odeslat a u kterých již došlo k vypršení platnosti, systém již znovu odesílat nebude. [30]

5.4 Zvolení technologie pro přenos

Nyní když jsou známy všechny technologie pro příjem zpráv, může nastat další otázka. Jaký typ technologie příjmu použít pro svou aplikaci? Vše záleží na způsobu zpracování dat.

Pokud data budou zpracovávána ihned po obdržení, a zároveň je k dispozici server s nepřetržitým chodem, je nejlepší volbou příjem pomocí protokolu HTTP. U tohoto protokolu je navíc významně nižší bezpečnostní riziko, než u následujících technologií.

Pokud se však data budou zpracovávat v pravidelných intervalech (např. jednou za týden) je dobrou volbou výběr FTP protokolu. Navíc je mnohem jednodušší sehnat tento typ serveru, jelikož slouží pouze pro ukládání souborů. Není tedy potřeba žádné složité instalace. Jedinou nevýhodou této technologie je její potencionální bezpečnostní riziko. FTP přenos není nikterak šifrovaný a snadno tak může dojít k odposlechnutí autorizačních údajů a dat.

SMTP protokol by měl být použit opravdu pouze v případě, že není možná žádná z předcházejících možností.

5.5 Zjištěné problémy

Během implementace jsem narazil na jeden velmi významný problém, který se v přijímaných datech často objevuje. Tímto problémem je číslování verzí událostí u dopravní

intenzity (viz výpis kódu 1), u klasických dopravních událostí je číslování verzí v pořádku. V dokumentaci je uvedeno, že každá další verze události zvýší verzi události o +1. Dále je uvedeno, že může nabývat hodnot od 1 až do 64565 [29].

Verzí tohoto typu událostí však přichází velké množství s častou frekvencí aktualizace. Proto je jasné, že výše uvedený rozsah hodnot rozhodně stačit nemůže. Je tak otázkou času, kdy tento rozsah bude vyčerpán.

NDIC poté u událostí zasílá číslo verze, které již bylo dříve u stejné události použito. U některých událostí se dokonce zdá, že číslování verzí nemá žádný systém. Například ráno je přijata verze události 6234 a večer stejná událost dorazí s číslem verze 6105. Je tedy potřeba, dát si na toto číslování verzí pozor a nespoléhat pouze na NDIC.

Nejlepším řešením je vytvořit své vlastní číslování, aby byla zaručena časová posloupnost těchto událostí.

```
<MSG id="1238" version="275" type="TL" planned="false">
  <MTIME format="YYYY-MM-DDThh:mm:ssTZD">
    <TGEN>2015-02-16T09:12:46+02:00</TGEN>
    <TSTA>2015-02-16T09:12:32+02:00</TSTA>
    <TSTO>2015-02-16T09:22:32+02:00</TSTO>
  </MTIME>
</MSG>

<MSG id="1238" version="270" type="TL" planned="false">
  <MTIME format="YYYY-MM-DDThh:mm:ssTZD">
    <TGEN>2015-02-16T09:10:13+02:00</TGEN>
    <TSTA>2015-02-16T09:10:05+02:00</TSTA>
    <TSTO>2015-02-16T09:20:05+02:00</TSTO>
  </MTIME>
</MSG>
```

Výpis 1: Zobrazení výstřižku XML zprávy s chybným číslováním verzí

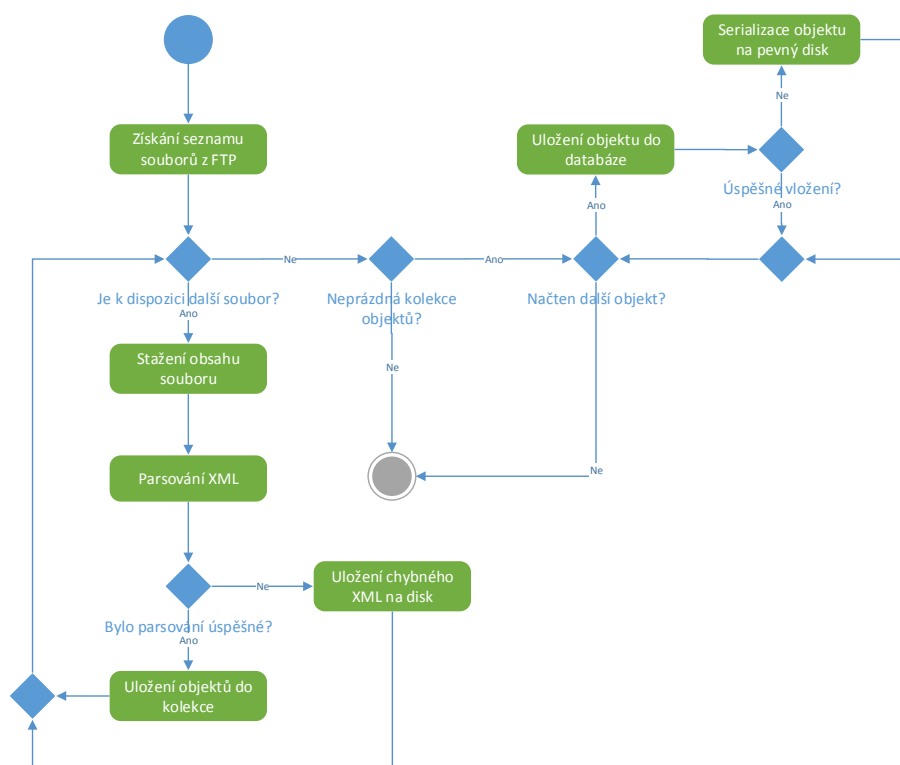
6 Implementace systému

6.1 Procesy v aplikaci

V této části implementace jsou zobrazeny některé základní diagramy aktivit. Pomocí tohoto typu diagramu lze zobrazit, k jakým procesům v aplikaci dochází a tedy lépe pochopit princip základních částí aplikace.

6.1.1 Získání a vložení dat

Tento diagram znázorňuje, jakým způsobem je vyřešeno získávání a ukládání dat. Na začátku procesu jsou získány názvy souborů uložené na FTP serveru. Pokud je na serveru nalezen alespoň jeden nebo více souborů, dojde k jejich stažení a následnému parsování. Během parsování jsou objekty ukládány do kolekce. Pokud jsou již uloženy objekty ze všech souborů, jsou tyto objekty postupně ukládány do databáze. V případě, že během ukládání konkrétního objektu došlo k chybě, je objekt uložen do chybové kolekce (viz kapitola 6.2.1) a následně serializován na pevný disk.

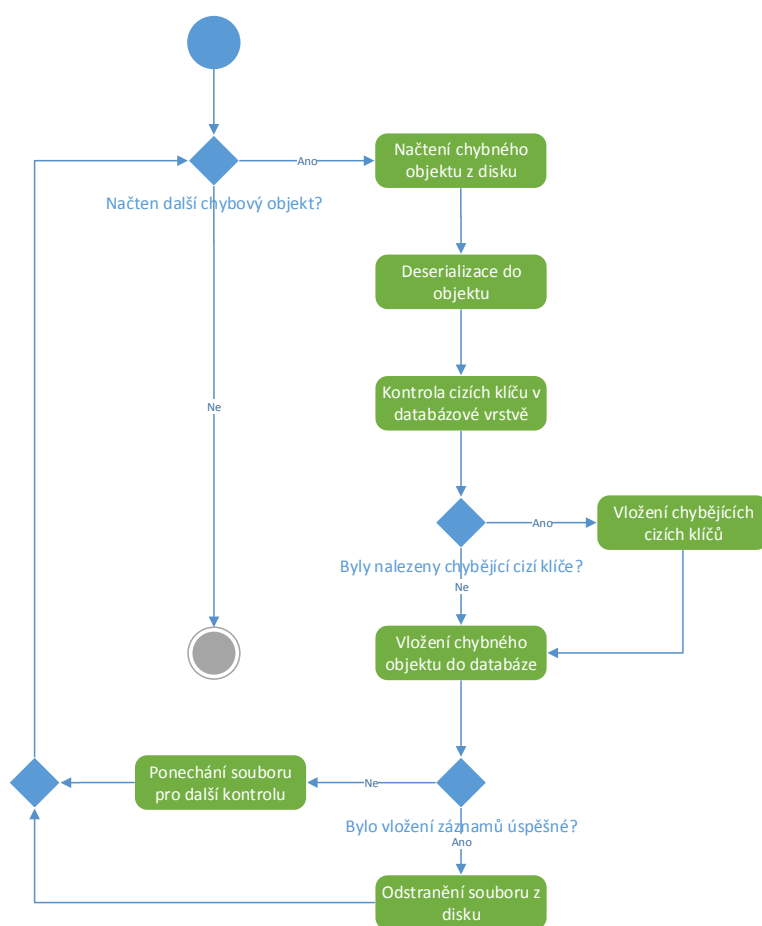


Obrázek 13: Diagram aktivity získávání a ukládání dat

6.1.2 Automatická oprava cizích klíčů

Tento proces znázorňuje, jakým způsobem dochází k automatické kontrole cizích klíčů. Pokud ukládání některých objektů do databáze selhalo, jsou tyto objekty vloženy do kolekce a následně serializovány na pevný disk (viz kapitola 6.2.1).

Pokud ve složce existuje alespoň jeden chybový objekt, dojde k jeho načtení a následné deserializaci do objektu typu `MessageData`. Poté je provedena kontrola cizích klíčů. Pokud jsou nalezeny záznamy, které nemají v databázi svůj cizí klíč, vloží tento klíč do databázové tabulky. Po této kontrole, i v případě že nebyly nalezeny žádné chyby, je proveden nový pokus o uložení dat. Pokud uložení proběhne v pořádku je soubor s objektem smazán z disku. V opačném případě je soubor ponechán na pevném disku pro další kontrolu. Tento proces je opakován, dokud nejsou otestovány všechny chybové objekty.

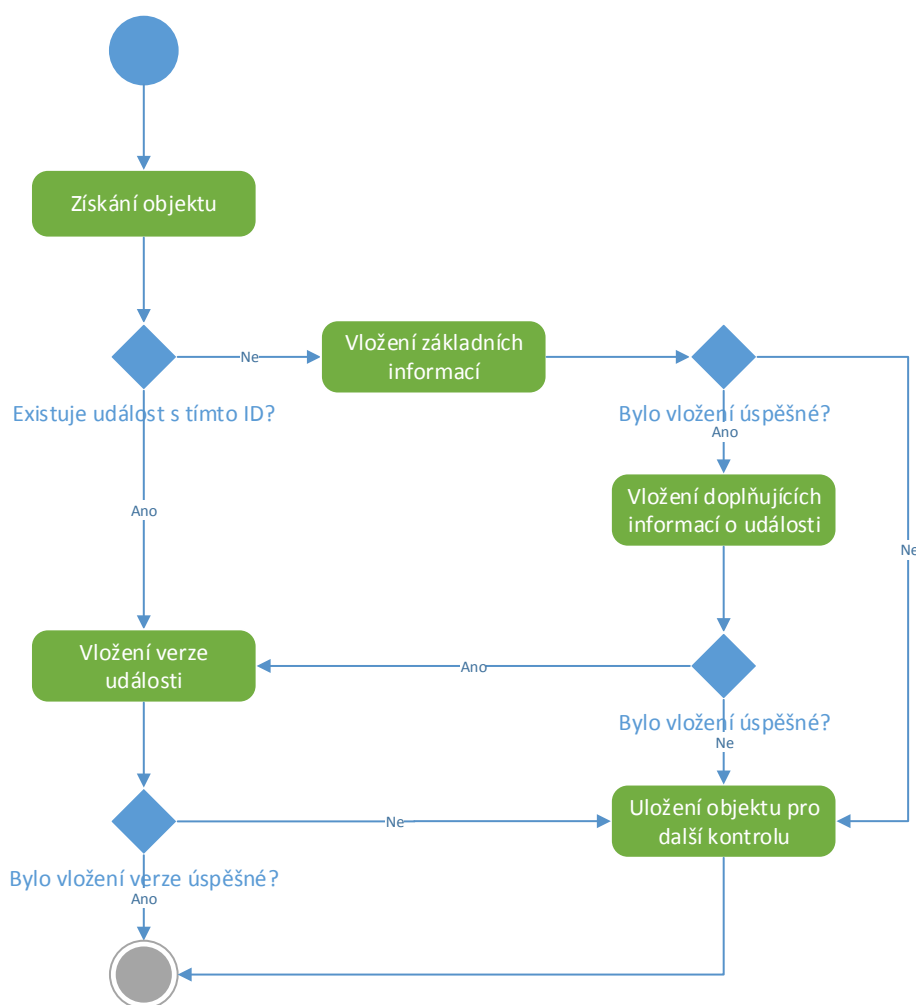


Obrázek 14: Diagram aktivity automatické kontroly cizích klíčů

6.1.3 Aktualizace dopravních dat

U tohoto procesu je zobrazeno, jakým způsobem dochází k aktualizaci dopravních informací. Diagram pro přehlednost zobrazuje aktualizaci dat pouze na jednom objektu.

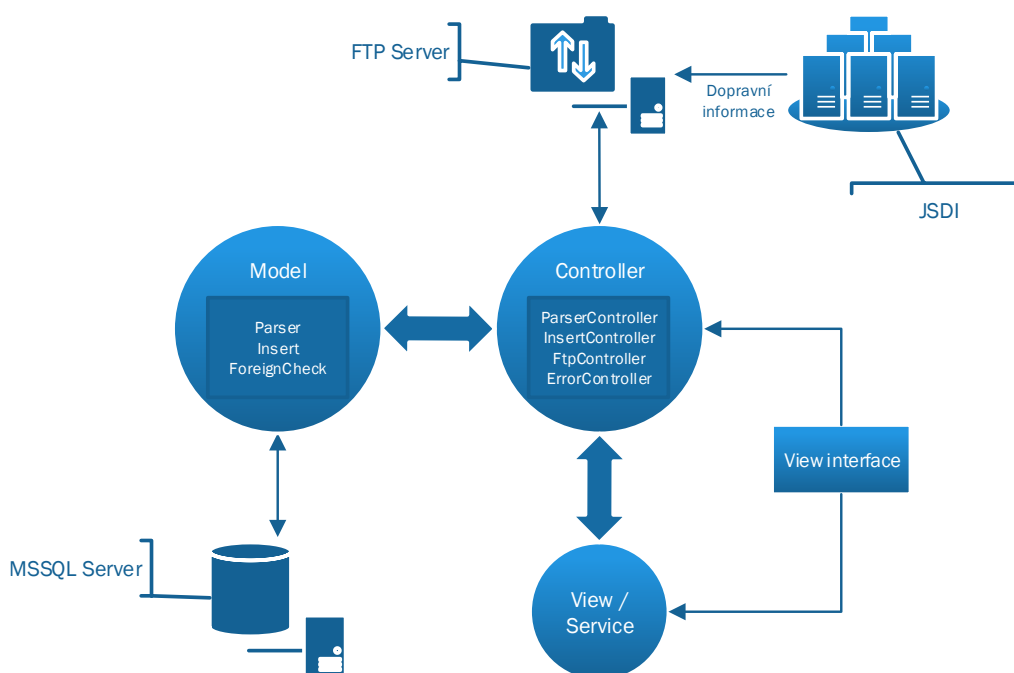
Po získání objektu typu `MessageData` je ověřeno, zda již v databázi existuje záznam se stejným jednoznačným identifikátorem. Pokud záznam v databázi ještě neexistuje, jsou vložena základní data o této události (viz tabulka 14). Dále jsou uloženy informace o ovlivněných úsecích pozemních komunikací a administrativních jednotkách. Nakonec je bez ohledu, zdali se jedná o záznam nový nebo již existující provedeno vložení verzovacích dat (viz tabulka 17).



Obrázek 15: Diagram aktualizace dat

6.2 Získávání dopravních informací

První celek je zodpovědný za získávání a zpracovávání dat. Tato část je naprogramována v programovacím jazyce C#. Při návrhu této části byl použit návrhový vzor MVC, který byl však mírně upraven. Celkem se tato část sestává ze dvou částí. První část byla psána jako Windows Form projekt a byl využíván především při vývoji parseru. Druhá část je napsána jako Windows služba, která automaticky vkládá nová data, případně kontroluje data chybná. Obě tyto části pracují nad stejnou logikou. Na obrázku 16 je nastíněno, jakým způsobem dochází v aplikaci k toku dat.



Obrázek 16: Tok dat u parsovací části

Zvolená metoda pro získávání dat

Jak je zmíněno v kapitole 5, dříve než bude možné získávat data, je nutné zvolit službu, pomocí které budou data přijímána. Pro tuto konkrétní implementaci je zvolen FTP protokol, a to hlavně z důvodu nedostupnosti webového serveru, který by pomocí HTTP protokolu dokázal data přijímat. Druhým důvodem, proč byl vybrán právě tento protokol, je způsob zpracování dat. Data totiž při začátku implementace byla zpracovávána přibližně jednou týdně, a právě pro tento způsob zpracování byl nejvíce vhodný FTP protokol.

6.2.1 Aplikační vrstva

Tato vrstva je zodpovědná za veškerou logiku aplikace. Získávají a připravují se v ní data pro modelovou vrstvu. Každý controller je jakousi mezivrstvou mezi prezentační a modelovou vrstvou. Z prezentační vrstvy, případně Windows služby, jsou data zaslána do controlleru, který data zpracuje. Po zpracování data zašle do modelové vrstvy, která data uloží. Po uložení controller zavolá prezentační vrstvu a vypíše výsledek operace.

ParserController

Tento controller má za úkol stáhnout a připravit data pro třídu `Parser`. Controlleru je předán počet, kolik souborů má z FTP stáhnout a připravit. Tento argument je poté předán `FtpControlleru`, konkrétně jeho metodě `ListFTP(limit)`, která vrátí kolekci s názvy souborů.

Po získání seznamu souborů je dalším krokem stažení dat z těchto souborů. Data jsou získávána paralelně pomocí třídy `ThreadPool` a její statické metodě `QueueUserWorkItem`, která je uvnitř cyklu procházející kolekci s názvy souborů. Jako argument je jí předán název metody pro obsluhu a název souboru pro stažení. V obsluhující metodě je provedeno stažení dat pomocí třídy `WebClient`. Získaná data jsou zaslány do modelové vrstvy, kde se dále zpracují.

InsertController

Z předchozího controlleru jsou data již naparsována a uložena v kolekci. Nyní je potřeba uložit tyto data do databáze. K přípravě dat pro uložení slouží právě `InsertController`. Získaná data jsou z `ParserControlleru` předána tomuto controlleru, který provede jejich seřazení podle verze a následné uložení do privátní kolekce. Poté je tato kolekce iterativně projeta a každý objekt typu `MessageData` je následně předán do modelové vrstvy, kde jsou data uložena do databáze. Před uložením dat je ověřeno, že na žádném jiném počítači neprobíhá jiné ukládání. Podrobnější popis je v kapitole 6.2.2.

FtpController

Jak již bylo výše zmíněno, jednou z funkcí `FtpControlleru` je získání kolekce s názvy souborů uloženými na FTP serveru. To je prováděno pomocí třídy `FtpWebRequest`, která zašle požadavek na server. V odpovědi jsou získány názvy všech uložených souborů, které se na serveru vyskytují. Ty jsou poté pomocí `StreamReaderu` přečteny a uloženy do kolekce.

Druhou funkci, kterou tento controller plní je mazání souborů z FTP, ať už po úspěšném či neúspěšném vložení dat. Aby však nedošlo ke kompletnímu vymazání všech souborů na FTP serveru, je nutné přesně specifikovat, které soubory se budou odstraňovat. `FtpController` si při získání názvu souborů uloží tuto kolekci do své privátní proměnné. Po dokončení ukládání tedy stačí smazat ze serveru pouze ty soubory, které jsou součástí této kolekce.

ErrorController

Tento controller, jak již název napovídá má na starosti práci s chybnými záznamy. Obsahuje metody pro udržování chybových zpráv vzniklých při parsování, případně ukládání. To je realizováno pomocí statické kolekce, do které se chybové zprávy ukládají. Nesmí se však zapomenout před každým dalším spuštěním kolekci promazat, aby nedocházelo k přenášení chybových zpráv, které již nejsou platné.

Další důležitou částí tohoto controlleru je metoda `Serialize`. Tato metoda slouží k uložení chybných objektů na disk. Tyto soubory mohou být poté opět pomocí metody `Deserialize` převedeny do objektů typu `MessageData`.

Tato funkce je používána i v případě automatické kontroly cizích klíčů. Pomocí této metody jsou data z disku načtena do kolekce objektů, nad kterými je poté iterativně zavolána metoda `Check` ze třídy `ForeignCheck`. Pokud tato třída opraví nějaké chybějící klíče, dojde k přejmenování souborů na disku, a to přidáním prefixu `FK-Checked`. Windows služba potom ví, že se může pokusit tyto soubory znovu vložit (více v kapitole 6.2.4).

6.2.2 Modelová vrstva

Tato vrstva je zodpovědná za získávání dopravních informací a jejich následné uložení do databáze. Hlavními stavebními prvky modelové vrstvy jsou třídy `Parser`, `Insert` a `ForeignCheck`. V této části je ukázáno, jak je tato vrstva navržena a jaké jsou její hlavní úkoly. Vrstva obsahuje i další třídu `SqlConnection`, která však jen vytváří a předává připojení k databázi.

Při práci s databází, je důležité dbát na to, aby se tato data nedostala do rukou nepovolaným osobám. Častým typem útoku, pomocí kterého se k těmto datům lze poměrně snadno dostat je SQL injection. Tento typ útoku spoléhá na podvržení SQL dotazu, který poté v databázi provede dotaz, který původně proveden být neměl. Řešením jak tento typ útoku neumožnit, je používání vázaných proměnných. V rámci celé aplikace jsou tedy proměnné hodnoty předávány pomocí vázaných proměnných.

Dalším důležitým bezpečnostním prvkem je zamezení vzdáleného přístupu k databázi. Databáze tedy nemá otevřený port pro vzdálený přístup. Lze k ní tedy přistoupit jen z lokálních skriptů a programů.

Parser

První třída, která je v rámci této vrstvy používána je třída `Parser`. Tato třída je zodpovědná za parsování a ukládání dat do objektu typu `MessageData`. Instance této třídy je vytvořena v `ParserControlleru`, který nad touto instancí zavolá metodu `ParseXMLData` a do parametru ji předá získaná data.

V této metodě je z těchto dat vytvořen objekt typu `XmlDocument`. Před samotným získáváním dat je ještě provedena validace získaného XML souboru. Validace je prováděna vůči XSD schématu, které bylo vytvořeno dle dokumentace (viz kapitola 5.2). Pokud je soubor označen jako validní, získají se pomocí jazyku XPath požadovaná data. Získaná data jsou následně uložena do objektu typu `MessageData`.

Protože `ParserController` stahuje a volá metodu `ParseXMLData` ve vláknech, je nutné vytvořené objekty ukládat do kolekce, která je schopna ukládat objekty v několika vláknech najednou bez rizika přepsání dat. Pro tyto účely byl použit typ kolekce `ConcurrentBag`, který je přímou součástí .NET Frameworku [31].

Insert

Tato druhá třída je zodpovědná za ukládání zpracovaných dat do relační databáze. Princip je opět velmi jednoduše pochopitelný. `InsertController`u jsou pomocí metody `Init` předána data, která se seřadí podle verze a uloží do privátní proměnné `controlleru`. Dále je v této metodě vytvořena nová instance třídy `Insert`. V této vytvořené instanci je provedena inicializace kolekce a jejich naplnění daty. Tyto kolekce jsou později využívány k rychlejšímu vkládání dat do databáze. V každé iteraci vkládání dat tedy odpadá nutnost provádět `SELECT` dotazy na získání potřebných dat pro cizí klíče.

V `InsertController`u je poté zavolána metoda `Insert`, ve které dochází k iterativnímu průchodu dříve zapsané kolekce se zpracovanými daty. Nad již vytvořenou instancí třídy `Insert` je v každé iteraci zavolána metoda `InsertNewItem`, které je už předán konkrétní objekt typu `MessageData`. Vkládání dat do databáze je ošetřeno pomocí databázových transakcí. V případě chyby tak není narušena konzistence dat.

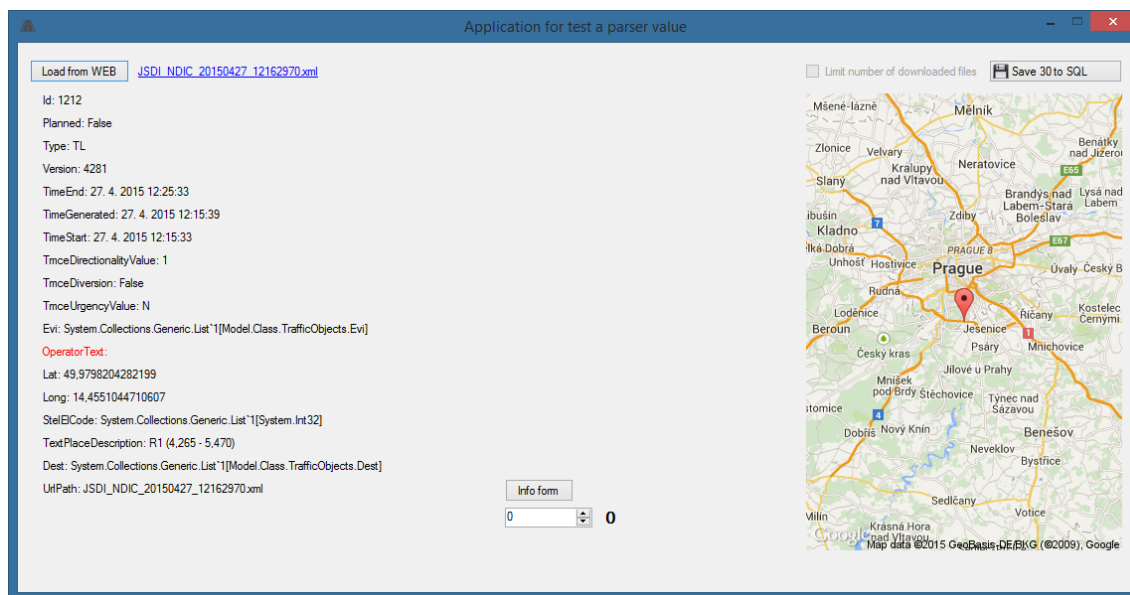
ForeingCheck

Poslední důležitou třídou je `ForeingCheck`, hlavním úkolem této třídy je automatické učení typů událostí, které ještě nejsou v databázi uloženy. Tato třída je volána pouze z vytvořené Windows služby (více v kapitole 6.2.4) a to pouze tehdy, jsou-li k dispozici nějaké objekty, u kterých nedošlo ke správnému vložení. `ErrorController` vytvoří instanci této třídy a iterativně ji předává objekty, u kterých došlo k chybě. Třída kontroluje, zdali existují všechny cizí klíče typů událostí, které má konkrétní objekt uložen. V případě že nalezne chybějící cizí klíč, dojde k jeho vložení. Tím je aplikace schopná automatického učení, v případě přidání nového typu události ze strany JSDI.

6.2.3 Prezentační vrstva

Prezentační vrstva původně vznikala jako jediná možnost pro ukládání dopravních informací. V dalších vývojových verzích ji však nahradila Windows služba (více v kapitole 6.2.4), která je pro opakované a zcela automatické vkládání dat vhodnější. Prezentační vrstva si však ponechala veškerou funkčnost, kterou disponuje i Windows služba. Tato vrstva se skládá ze dvou hlavních oken.

Hlavní okno aplikace (viz obrázek 17) není nikterak složité. Obsahuje pouze několik málo komponent sloužících k ovládání aplikace. V levé části je vložen `Panel`, který v přehledné formě zobrazuje veškeré naparsované informace. Pravé části dominuje velká mapa, která zobrazuje výskyt události. Nahoře jsou pak pouze tlačítka pro získání dat a jejich následné uložení. Po získání dat okno umožňuje listování a případné kontrolování získaných dopravních informací.



Obrázek 17: Okno se získanými informacemi

Další okno slouží pro zobrazení chyb. Toto okno je vyvoláno pouze v případě, že během získávání dat, případně během jejich ukládání došlo k chybě. V případě vzniku této události jsou informace o chybě přehledně zobrazeny právě pomocí tohoto okna. Pro přehlednost byla ukázka tohoto okna (viz obrázek 37) zařazena do příloh.

6.2.4 Windows služba

Vznikla jako náhrada prezentační vrstvy popsané v kapitole 6.2.3. Původním plánem bylo spouštění parsování a vkládání dat pouze z prezentační vrstvy.

Postupem času však přišla chvíle, kdy bylo nutné celý proces zautomatizovat. Existovaly však dvě možnosti, jak této automatizace dosáhnout. Jednodušším a rychleji realizovatelnějším způsobem by bylo použití `Timeru` v prezentační vrstvě. Druhou možností bylo vytvoření Windows služby, která se pro takové situace hodí mnohem lépe. Nakonec byla zvolena právě Windows služba.

Velkou výhodou při vlastní implementaci bylo dodržování návrhového vzoru MVC. Odpadlo tak složité psaní nové logiky a stačilo tak pouze služby napojit na stávající aplikační vrstvu (viz kapitola 6.2.1). Součástí této vrstvy je jedna Windows služba, která však obsahuje další dvě služby. Ukázku, jak lze tyto služby spustit zobrazuje výpis zdrojového kódu 2.

```
private static void Main()
{
    TrafficParseService service = new TrafficParseService();
    ForeignKeyCheckService fknService = new ForeignKeyCheckService();

    ServiceBase[] ServicesToRun = new ServiceBase[]
    {
        service,
        fknService
    };

    ServiceBase.Run(ServicesToRun);
}
```

Výpis 2: Spuštění několika služeb současně

Aby však bylo zajištěno, že vždy bude aktivní pouze jedna operace v rámci služby, bylo důležité vymyslet kontrolu, která toto zajistí. Pro tuto kontrolu byla vytvořena statická třída `ProcessCheck`. Tato třída obsahuje dvě statické proměnné, první určuje, zdali je nějaký proces aktuálně spuštěný. Při spuštění služby je tato proměnná nastavena na hodnotu umožňující spuštění operace. Před každým spuštěním operace je pak otestováno, že žádná operace není aktuálně spuštěna. Pokud je tato podmínka splněna, operace je spuštěna a proměnná je nastavena tak, že neumožní spuštění další operace, dokud neskončí aktuální. Po dokončení aktuální je proměnná opět nastavena tak, aby umožnila další spuštění. Druhá proměnná slouží pouze pro uložení názvu operace, která je aktuálně spuštěna. Tato hodnota je potom zapisována do prohlížeče událostí (viz obrázek 38).

Parsovací služba

Prvním službou v rámci této vrstvy je `TrafficParseService`. Jak již název napovídá, tato služba je zodpovědná za automatické parsování informací a následné uložení dat do databáze.

Automatické provádění je realizováno pomocí `Timeru`, který každé tři minuty provede stažení přesného počtu souborů, jejich parsování a následné uložení (viz kapitola 6.2.1). Pokud by však tento proces trval příliš dlouho, mohlo by dojít ke spuštění dalšího ukládání dat. Bylo tedy důležité vymyslet mechanismus, který bude kontrolovat, zdali je původní ukládání již dokončeno a neběží žádná další operace. Tento mechanismus je popsán v kapitole 6.2.4.

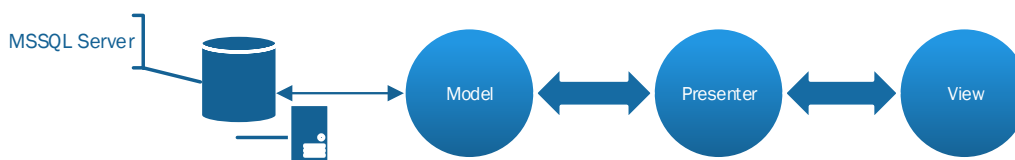
Kontrolovací služba

Další služba je pojmenována `ForeignKeyCheckService`. Úkolem této služby je automatická kontrola cizích klíčů u záznamů, které se nepodařilo úspěšně uložit službě předchozí.

Automatizace je opět prováděna pomocí třídy `Timer`. V tomto případě však není nastaven pevný interval, ve kterém se kontrola spouští. Tento interval je vytvářen zcela náhodně, a to z důvodu, aby byla co nejvíce minimalizována možnost spuštění obou služeb ve stejný čas. Pokud by však takováto situace nastala, aplikace povolí spuštění pouze první operace (viz kapitola 6.2.4).

6.3 Webová vrstva

Webová část projektu je zodpovědná za vizualizaci získaných dat ve webovém prohlížeči. V této části kapitoly je nastíněna příprava dat, použité technologie apod. Možnosti webové aplikace jsou poté popsány v kapitole 6.4.



Obrázek 18: Tok dat u webové části

6.3.1 Použité technologie

Webových frameworků a technologií existuje velké množství. Z takového množství technologií nemusí být jednoduché vybrat ty správné. Zpočátku jsem zvažoval, že by tato část byla napsána v jazyku PHP bez jakéhokoliv frameworku. Po delší úvaze jsem došel k názoru, že by to nebyla úplně nejlepší volba. Bylo potřeba tedy vybrat nějaký framework, který by při vývoji nedovolil psát špatně udržitelný a nepřehledný kód. Rozhodnutí tedy velmi brzy padlo na Nette Framework. Tento framework disponuje vyzrálým objektovým návrhem, vysokým výkonem a výborným šablonovacím systémem Latte. Nette Framework je postaven na návrhovém vzoru MVP (Model - View - Presenter) [32], který vývojáře vede k jeho správnému používání.

Technologie pro zpracování skriptů na straně serveru je již známá, pro skripty vykonávané na straně klienta jsou využívány následující knihovny:

- jQuery,
- Bootstrap 3,

- Google Maps API v3,
- Google Chart API.

6.3.2 Moduly

Webová aplikace je pro čistější a lépe udržitelný kód rozdělena do dvou samostatných modulů. Každý modul obsahuje vlastní `Presentery` a `Modely`. Prvním je základní modul, který je volán v případě, že se jedná o klasického návštěvníka aplikace. Druhým je JSON modul, ten slouží pro obsluhu AJAX požadavků.

Základní modul

Jak již bylo výše zmíněno, tento modul je používán vždy v případě, když se uživatel pohybuje po webové stránce. Neobsahuje žádnou významnou logiku aplikace. Obsahuje pouze definici vzhledu celé aplikace a vytvoření několik komponent pro aplikaci filtrů.

K šabloně jsou připojeny Javascriptové soubory, které jsou zodpovědné za veškeré fungování filtrů, analýz i statistik. Veškerá volání na JSON modul jsou prováděna pomocí technologie AJAX, není tedy nutné při tomto požadavku obnovovat stránku. Tato technika výrazně zrychlí reakce webového serveru, jelikož není nutné přenášet znovu celý obsah stránky. Jsou přenášeny pouze změněné fragmenty stránky [33].

Nastavení uživatelských filtrů jsou ukládána do `cookies` webového prohlížeče a jejich platnost je nastavena na 14 dnů. Po uzavření a opětovném otevření prohlížeče budou tedy veškeré filtry nastavené tak, jako před opuštěním stránky. U jednodušších filtrů, které nelze nijak kombinovat jsou data zasílána přímo pomocí `GET` parametrů.

JSON modul

Tento modul naopak od základního nemá definován žádný vzhled. Jeho hlavním úkolem je vrátit Javascriptu zpracovaná data ve formátu JSON. Tento formát je v současné době nejrozšířenější právě pro tento způsob využití. Postupem času nahradil XML formát, který má při stejném obsahu dat několikanásobně větší velikost [34]. Na příkladu 6.1 je uvedeno, jak je získávání dat v Nette Frameworku realizováno.

Příklad 6.1

V Javascriptu je pomocí technologie AJAX vytvořen požadavek na webovou adresu (například `/json/analytics/?km=5`). Nastavení routerů rozpozná, že má požadavek směřovat do JSON modulu na presenter `AnalyticsPresenter` a v něm zavolat metodu `actionDefault`. Tento presenter si mezitím z DI kontejneru získá instanci modelu, se kterým bude dále pracovat. V metodě `actionDefault` už jsou pouze získána a zpracována data, která jsou pomocí `payload` (viz zdrojový kód 3) zaslána zpět Javascriptu. Ten data dále zpracuje a zobrazí.

```
class AnalyticsPresenter extends BaseDataPresenter
{
    /** @var Model\DataAnalyticsRepository @inject */
    public $analyticsRepository;

    public function actionDefault($km)
    {
        $this->payload->data = $this->analyticsRepository->getAnalyticsData($km);
        $this->sendPayload();
    }
}
```

Výpis 3: Ukázka zaslání payloadu z presenteru

■

Podobných presenterů obsahuje tento modul několik, některé jsou zodpovědné za zaslání dat pro mapu, jiné za vytváření dat pro analýzu a statistiku. Všechny jsou popsány ve vygenerované dokumentaci přiložené na DVD.

6.3.3 Optimalizace získávání dat

Při práci s daty, která jsou často měněna, je vždy nutné dbát na to, aby uživatel pracoval s daty aktuálními. Vývojář zase potřebuje, aby uživatel tyto data dostal co nejrychleji a pokud možno s co nejnižším zatížením serveru.

Nette Framework obsahuje systém mezipaměti, která je pro tyto situace jako dělaná. Mezipaměť je automaticky vymazána v případě, že došlo ke změně v souboru, na kterém je mezipaměť závislá. Každá tabulka databáze má tedy vytvořený samostatný `*.neon`

soubor, který obsahuje pouze informaci o počtu záznamů. Pokud se počet záznamů nezměnil, dojde k načtení dat z mezipaměti. V opačném případě jsou data načtena z databáze a opět uložena. Aby však bylo zajištěno, že kontrola dat v databázi nebude prováděna při každém požadavku, bylo nutné vyřešit i tento problém. Při prvním získávání počtu dat tabulky je vytvořen soubor, který je pojmenován stejně jako tabulka. Při dalším požadavku jsou přečteny atributy tohoto souboru, a pokud byl změněn před méně, než definovaným časem, dotaz na počet záznamů se neprovádí. V opačném případě je dotaz proveden a u souboru je změněn atribut změny souboru na aktuální datum a čas. Zjednodušená ukázka tohoto principu je zobrazena ve zdrojovém kódu 5. Tato část kódu vždy zajistí, že zobrazovaná data budou vždy aktuální a přitom se rychlost získávání dat několikanásobně zvýší.

V ukázce zdrojového kódu 4 je zobrazeno jak tuto inteligentní mezipaměť nastavit.

```
/** @var App\Classes\Nette\MyCache */
private $cache;

public function getCacheData {
    return $this->cache->load(__METHOD__, function (& $dp) {
        $dp = [
            App\Classes\Nette\MyCache::FILES => "TrafficData.neon",
        ];

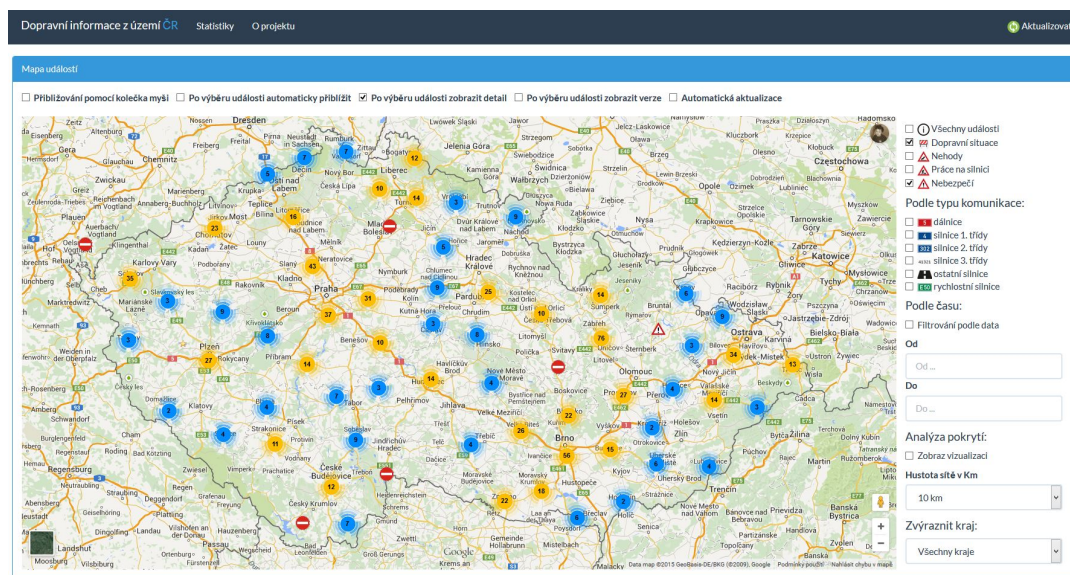
        $data = $this->getSomeData();
        return $data;
    });
}
```

Výpis 4: Automatická invalidace cache při změně souboru

6.4 Webová aplikace

V předchozí části kapitoly 6.3 je nastíněn princip funkčnosti a přípravy dat pro jejich následné zobrazení. Tato podkapitola je již věnována možnostem, které webová aplikace uživatelům umožňuje.

Celá webová aplikace obsahuje pouze jednu hlavní stránku, které obsahuje mapu se zobrazenými informacemi v horní části stránky. Vedle mapy je velké množství filtrů, podle kterých je možné filtrovat zobrazené informace na mapě (viz obrázek 19). Dalším výrazným prvkem na stránce je sekce se statistikami. Webová aplikace je dostupná na adrese <http://www.dopravni-info.eu>.








Obrázek 19: Ukázka zobrazení webové stránky

6.4.1 Filtrování událostí

Jak bylo zmíněno v kapitole 6.4, u mapy jsou zobrazeny filtry, pomocí kterých lze snadno nakonfigurovat události, které budou v mapě zobrazeny. Tato část kapitoly je věnována možnostem konfigurace jednotlivých filtrů. Po změně nastavení filtrů, nedojde ve většině případů k automatickému obnovení událostí, nýbrž je nutné potvrdit tyto nastavení tlačítkem **Filtrovat**. Konfiguraci filtrů lze mezi sebou libovolně kombinovat.

Typ událostí

Pomocí tohoto filtru lze snadno zobrazit pouze určitý typ události. Po prvním načtení jsou zobrazeny všechny dostupné události bez rozdílu typu. Po změně na určitý typ je v mapě zobrazen pouze tento vybraný typ. Samozřejmostí je výběr více druhů, které lze libovolně kombinovat.

- ☒  Všechny události
- ☐  Dopravní situace
- ☐  Nehody
- ☐  Práce na silnici
- ☐  Nebezpečí

Obrázek 20: Výběr typu události

Typ pozemní komunikace

Tento filtr získává data v závislosti na typu zvolené pozemní komunikace. Pokud není vybrán žádný typ komunikace, jsou použity všechny typy. Naopak při výběru určitého typu jsou zobrazeny pouze události na vybraném typu komunikace. Musí však být splněna podmínka, že NDIC u dopravních informací vyplnil údaje o typu komunikace. V případě, že tyto informace u některých událostí nevyplnil, nedojde k zobrazení těchto událostí na mapě. Lze vybrat více typů komunikací, které lze libovolně kombinovat.

- ☒  **dálnice**
- ☐  **silnice 1. třídy**
- ☐  **silnice 2. třídy**
- ☐  **silnice 3. třídy**
- ☐  **ostatní silnice**
- ☒  **rychlostní silnice**

Obrázek 21: Výběr události podle typu pozemní komunikace

Výběr kraje

Úkolem tohoto filtru je získání dopravních informací pouze pro vybraný kraj. Po výběru kraje jsou tedy získány pouze události, které mají svou polohu ve vybraném kraji. Po získání dat je vybraný kraj automaticky přiblížen a jsou zvýrazněny jeho hranice. Pokud je vybrána položka *Všechny kraje* je mapa automaticky oddálena a opět zobrazeny události z celého území republiky.



Obrázek 22: Zvýraznění kraje

Výběr data a času

Další filtr umožňuje vybrat události podle zadaného časového rozmezí. Po povolení a zadání obou údajů jsou získány dopravní informace, které byly v tomto časovém úseku vytvořeny. V případě, že filtrování podle data není povoleno, nebo není zadán některý z časových úseků, zobrazí se aktuální dopravní informace.

☐ Filtrování podle data

12.03.2015 10:36

27.03.2015 10:37

◀ ▶

↑ ↓

Březen 2015

Po	Út	St	Čt	Pá	So	Ne	
23	24	25	26	27	28	1	10:00
2	3	4	5	6	7	8	11:00
9	10	11	12	13	14	15	12:00
16	17	18	19	20	21	22	13:00
23	24	25	26	27	28	29	14:00
30	31	1	2	3	4	5	15:00

Obrázek 23: Výběr události podle data

Analýza dat

V kapitole 7.1, která se věnuje analýze dat na území ČR je zmínka o velikosti vybraného území. Tento filtr umožňuje výběr zmíněné velikosti území, pro které budou data získávána. Dále umožňuje povolit, případně zakázat zobrazení analýzy v mapě. Filtr přebírá nastavení ze všech ostatních filtrů. Nespolupracuje však s filtry výběru typu komunikace a kraje.

☐ Zobraz vizualizaci

Hustota sítě v Km

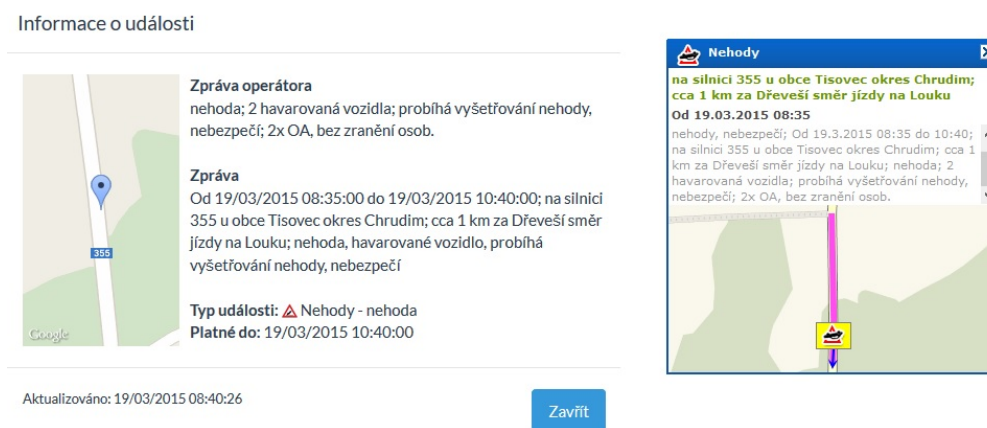
5 km

▼

Obrázek 24: Analýza událostí

6.4.2 Zobrazení detailu události

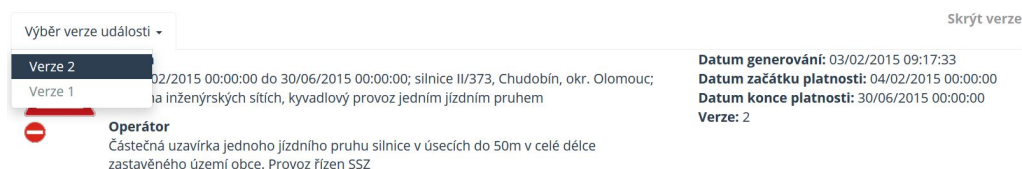
Další funkcí webové aplikace je zobrazení detailu vybrané události. Po výběru události je zobrazeno okno (viz obrázek 25), které zobrazuje detailní informace o této události. Levá část obrázku zobrazuje informace zobrazené touto aplikací. V pravé části obrázku lze vidět detail události, jak je interpretován na webu JSDI ⁵. Dále si lze povšimnout, že JSDI u události zobrazuje i úsek, který je touto událostí ovlivněn. Tuto funkcionalitu nebylo možné z důvodu absence lokačních tabulek pozemních komunikací (viz kapitola 3.2) vizualizovat.



Obrázek 25: Porovnání zobrazení detailu události

6.4.3 Zobrazení verzí události

Aplikace dále umožňuje zobrazit (viz obrázek 26) posledních 20 verzí u každé události. Tato funkcionalita je vhodná pro porovnávání rozdílů mezi jednotlivými verzemi události. U některých typů dopravních informací tak lze přehledně pozorovat, jak se informace o události postupně upřesňují. To se týká především dopravních nehod, nebezpečných situací a prací na vozovce.



Obrázek 26: Zobrazení verzí události

⁵<http://mapa.dopravniinfo.cz/>

7 Analýza a statistiky

7.1 Analýza četnosti událostí

Tato kapitola je věnována vizualizacím znázorňujících četnost zasílaných dopravních informací. Tato analýza je prováděna nad územím České republiky, které je rozděleno do oblastí stejné velikosti. V první části kapitoly je nastíněn výpočet oblastí. Další část se věnuje již samotným analýzám.

7.1.1 Výpočet oblastí

Velikost oblastí, pro kterou jsou data získávána, závisí zcela na výběru uživatele. Ten může vybrat z rozměrů o velikosti 3 x 3 km, 5 x 5 km, 10 x 10 km, 20 x 20 km a 50 x 50 km (viz kapitola 6.4.1). Po tomto výběru je nutné republiku rozdělit do stejně velkých oblastí. Výpočet oblastí však není tak jednoduchý, jak by se mohlo zdát. Ze zeměpisných souřadnic totiž nelze bez dalších výpočtů zjistit délku linie. Následující část je věnována výpočtům, pomocí kterých lze zjistit, kolik stupňů zeměpisné šířky a délky je potřeba k vyobrazení 1 kilometru na mapě.

Zeměpisná délka

Pro výpočet 1 km zeměpisné délky je potřeba znát pouze souřadnici zeměpisné šířky a obvod Země na rovníku. I když se předchozí věta může zdát jako nesmysl, není tomu tak. Obvod Země se směrem od rovníku zmenšuje. Pokud by tedy nebyla zadána zeměpisná šířka, zjistilo by se, kolik stupňů zeměpisné délky je 1 km pouze na rovníku. Na příkladu 7.1 je ukázáno, jak by výpočet probíhal na území České republiky.

Příklad 7.1

Než bude možné začít samotný výpočet, je důležité si ujasnit, kterou ze souřadnic zeměpisné šířky použít. Nejjednodušším řešením je zjistit nejjižnější (48.551802) a nejsevernější (51.055720) bod území, nad kterým se bude výpočet provádět. Z těchto získaných hodnot se průměrem vypočítá zeměpisná šířka, která bude dále ve výpočtech používána. Tento postup však lze použít pouze na území, která nejsou příliš rozsáhlá.

- Do proměnné *prumSirka* je uložena průměrná hodnota z

$$(48.551802 + 51.055720)/2 = 49.803761$$

- Do další proměnné *obvodZeme* je uložena hodnota 40075.017, což je hodnota obvodu Země na rovníku ⁶
- Dále vytvoříme proměnnou *obvodZemeCz*, která později bude sloužit pro uložení mezi výpočtu

⁶<http://en.wikipedia.org/wiki/Earth>

Nyní jsou již nadefinovány všechny proměnné a lze přejít k samotnému výpočtu. V prvním kroku výpočtu je důležité zjistit obvod Země na výše vypočítaných souřadnicích zeměpisné šířky. Matematický zápis by tedy byl [35]:

$$\begin{aligned} obvodZemeCz &= \cos(prumSirka) * obvodZeme \\ obvodZemeCz &= \cos(49.803761) * 40075.017 \\ obvodZemeCz &= 0,645408 * 40075.017 \\ obvodZemeCz &= 25864.736572 \end{aligned}$$

Posledním krokem výpočtu, je vydělení čísla 360 s právě získaným výsledkem. Tímto výpočtem získáme hodnotu, reprezentující kolik stupňů zeměpisné délky je potřeba pro zobrazení 1 km v České republice. Samotný výpočet je tedy:

$$\begin{aligned} x &= 360 / obvodZemeCz \\ x &= 360 / 25864.736572 \\ x &= 0,013919 \end{aligned}$$

■

Zeměpisná šířka

Výpočet, kolik stupňů zeměpisné šířky je potřeba pro zobrazení 1 km na mapě, je o něco jednodušší, než předchozí výpočet. K tomuto výpočtu je totiž potřeba znát pouze obvod Země přes póly. Tato hodnota je přibližně 40007.86 km⁷.

Nyní se pouhým vydělením čísla 360 s deklarovaným obvodem vypočítá, kolik stupňů zeměpisné šířky je potřeba pro zobrazení 1 km na mapě. Obvod Země přes póly je na všech místech Země totožný, není tedy potřeba znát žádné jiné údaje jako u výpočtu zeměpisné délky. Následující vzorec zobrazuje výpočet:

$$\begin{aligned} x &= 360 / obvodPresPoly \\ x &= 360 / 40007.86 \\ x &= 0,008998 \end{aligned}$$

7.1.2 Analýza

Tato část je již věnována samotné analýze četnosti zasílání dat na našem území. Princip analýzy je vcelku jednoduchý. Česká republika je rozdělena do oblastí stejné velikosti. Nad každou z těchto oblastí je poté dle zadaných filtrů (viz kapitola 6.4.1) proveden dotaz do databáze, pomocí kterého je zjištěn počet událostí v dané oblasti. Každá oblast, může být barevně označena jednou z desíti barev, která znázorňuje četnost výskytu informací. Čím více má oblast sytější odstín, tím více obsahuje událostí. Lze tedy jednoduše zjistit, které části republiky obsahují nejvíce, případně nejméně událostí.

⁷<http://en.wikipedia.org/wiki/Earth>

Princip výpočtu barevného označení

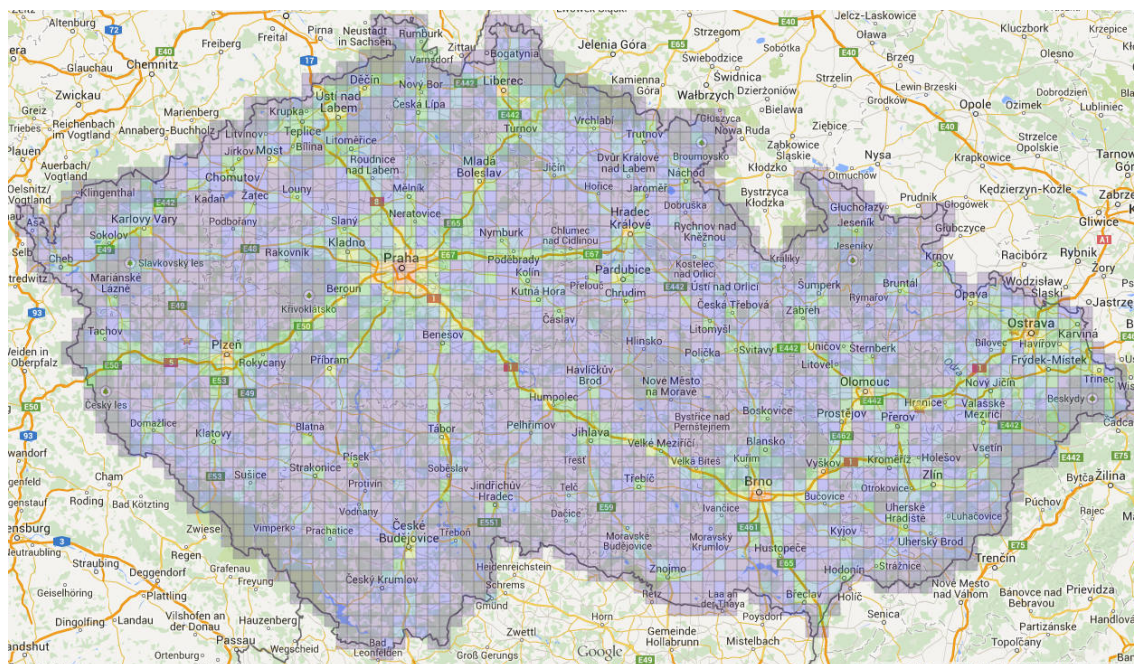
Pro toto označování částí, je používáno logaritmické měřítko s variabilním základem. To znamená, že je vždy dopočítáván, jaký základ logaritmu bude pro dané zobrazení použit. Tento systém je tedy vhodný pro jakékoliv množství záznamů.

Výpočet základu je prováděn tak, že po získání všech částí je nalezena část s největším počtem událostí. Tato část je poté definována jako referenční. Jak bylo již výše zmíněno, každá část může být barevně označena jednou z desíti barev. V programovacích jazycích budou tedy barvy uloženy v poli s indexy 0, 1, 2, ..., 9. Část s největším výskytem událostí by logicky měla obsahovat nejsytlejší barvu. Pomocí iterativního přírůstku je tedy hledán základ logaritmu (viz zdrojový kód 6), pro který bude mít maximální zjištěná hodnota výsledek 9. Tento základ je poté použit pro vykreslení i v ostatních částech.

Po vysvětlení, na jakém principu je obarvování částí založeno, je na čase ukázat si pár zajímavých analýz.

Výskyt všech událostí

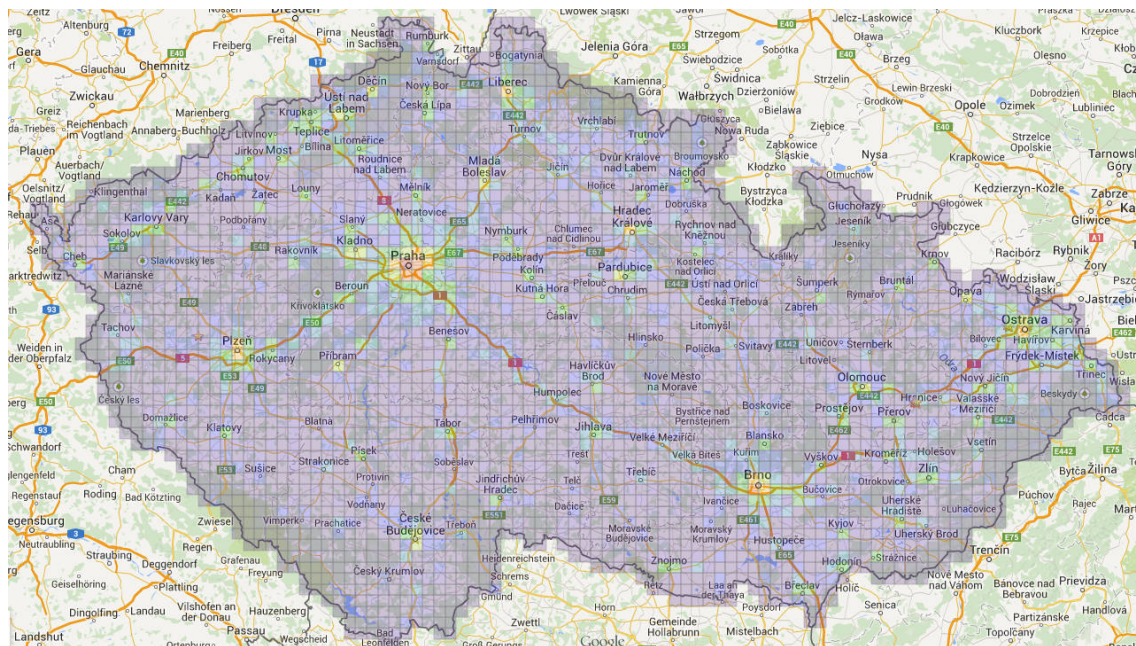
První analýza se bude zabývat výskytem dosud získaných dat bez rozdílu typu. Jak lze vidět na obrázku 27 největší četnost událostí je v oblasti tří největších měst naší republiky. Těsně v závěsu jsou české dálnice a silnice 1. tříd. Naopak u silnic nižších tříd je výskyt událostí spíše ojedinělý.



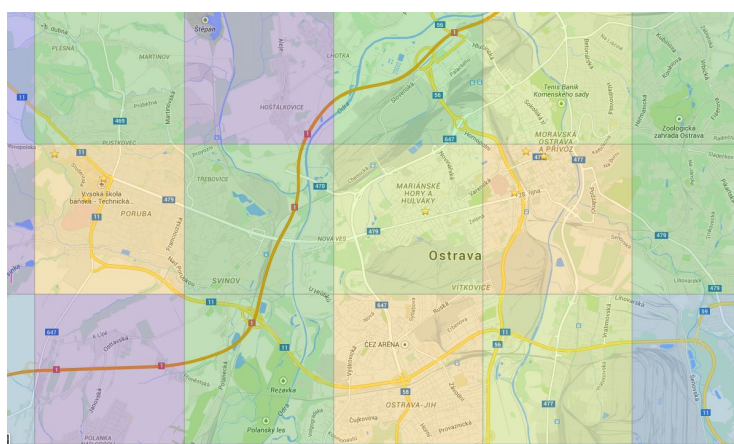
Obrázek 27: Četnost událostí pro všechny typy s velikostí čtverce 5 km

Výskyt nehod

U nehod je situace velmi podobná jako v předcházejícím případě. Největší počet dopravních informací typu dopravní nehoda přichází z největších měst naší republiky. Zvýšený výskyt nehod je i v některých dalších menších městech. U dálnic je naopak výskyt nehod velmi nízký.



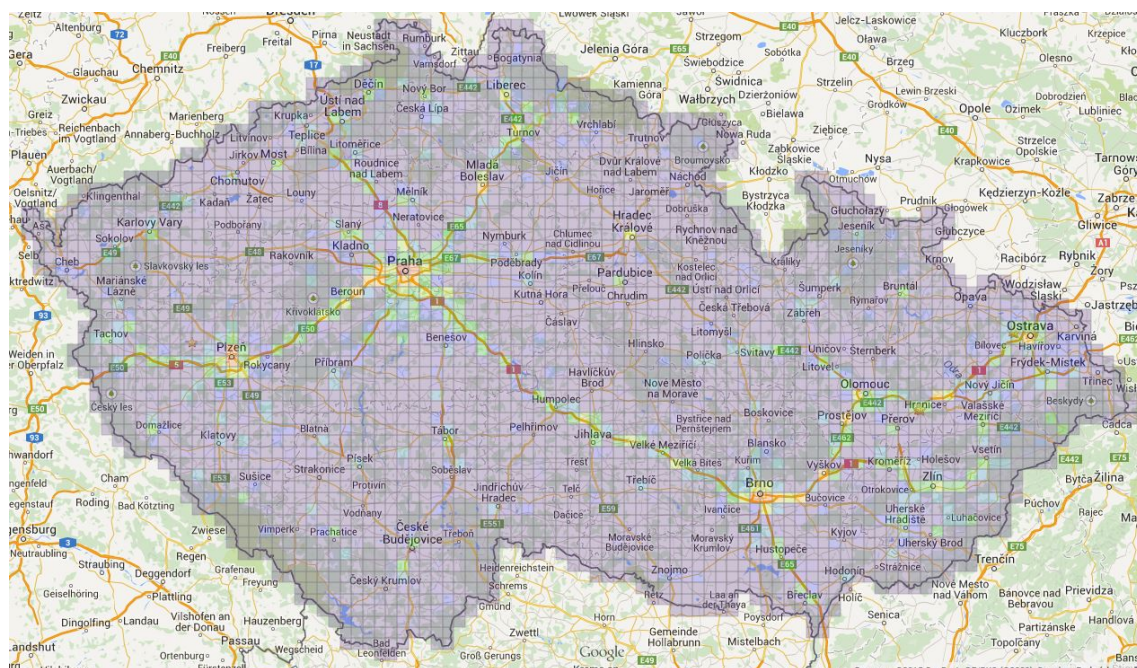
Obrázek 28: Četnost nehod s velikostí čtverce 5 km



Obrázek 29: Četnost nehod v Ostravě, čtverec 3 km

Výskyt nebezpečných situací

Nebezpečné situace se stejně jako u předchozích typů nejčastěji vyskytují ve všech třech největších městech. Velké množství nebezpečných situací je také přes velkou část dálnice D1. Vyšší počet je také u některých komunikací první třídy, především pak v okolí Prahy, Zlínska a Mladé Boleslavi. U tříd nižších je výskyt nebezpečných situací spíše ojedinělý.



Obrázek 30: Četnost nebezpečných situací s velikostí čtverce 5 km

Veškeré analýzy včetně možností aplikace jednotlivých filtrů (viz kapitola 6.4.1) jsou dostupné ve webové prezentaci popsané v kapitole 6.4. Ta navíc umožňuje po kliknutí na libovolnou část zjistit, kolik událostí v dané části bylo vytvořeno.

7.2 Statistiky dopravních informací

V kapitole 7.1 jsou zobrazeny analýzy nad získanými daty z pohledu četností výskytu podle území. Tato kapitola se věnuje některým zajímavým grafům a vizualizacím získaných z dosud uložených dat. Součástí této kapitoly jsou pouze některé zajímavé grafy. Všechny grafy jsou pak dostupné na webové části projektu (viz kapitola 6.4). V průběhu sběru dat, tedy od 12. 12. 2014 do 26. 04. 2015 bylo vytvořeno celkem **46 059** unikátních událostí. K těmto událostem bylo uloženo **5 942 411** verzí.

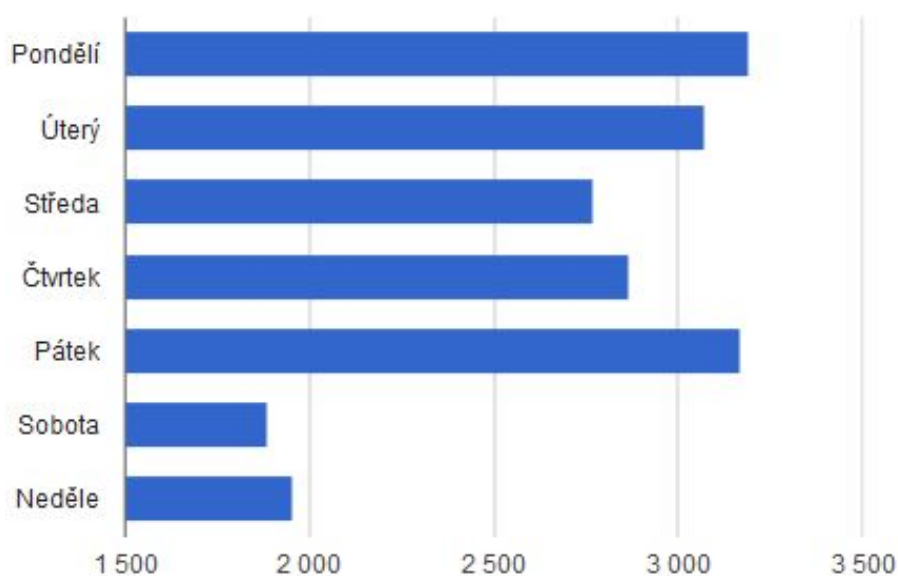
Výběr dat

Pro vytváření statistik jsou využívána reálná data, která byla přijata a zpracována pomocí vytvořené aplikace popsané v kapitole 6.2. Při vytváření grafů, byla použita všechna dostupná data a nebyla prováděna žádná odlehlá pozorování, které by vyloučilo data u kterých je hodnota velmi odlišná od hodnot ostatních.

Pokud by však byly v budoucnu nad těmito daty prováděny pokročilejší statistické operace, bylo by důležité tato pozorování provést, aby se zamezilo zbytečnému ovlivňování dat ostatních.

7.2.1 Výskyt nehod podle dnů

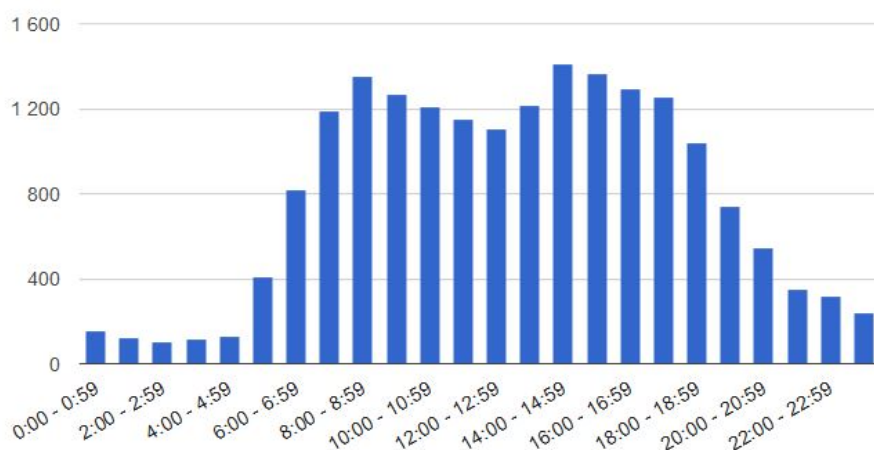
První statistický graf zobrazuje výskyt nehod podle dnů v týdnu. Jak se dalo předpokládat, pracovní dny jsou na nehody mnohem početnější, než dny víkendové. S počtem dopravních nehod je na tom nejhůře pondělí a pátek. Naopak nejméně dopravních nehod je v sobotu a neděli.



Obrázek 31: Počet nehod podle dnů v týdnu

7.2.2 Výskyt nehod podle hodin

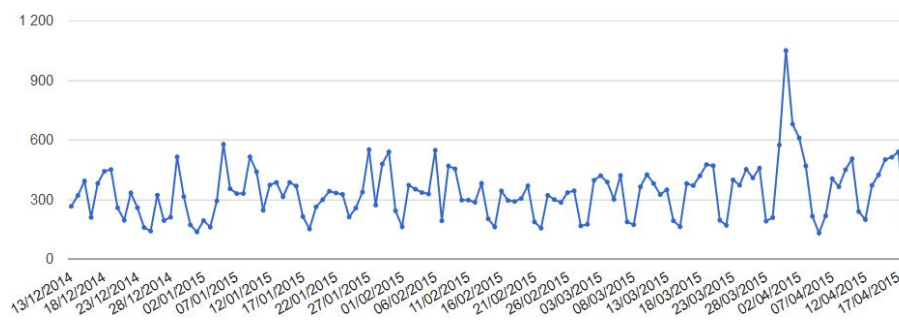
Další z grafů se opět týká dopravních nehod. Tentokrát však zobrazuje vznik dopravních nehod podle denní hodiny. Jak je z grafu (viz obrázek 32) již na první pohled patrné, nejvíce nehod vzniká od šesté do dvacáté hodiny. Nejnižší riziko dopravní nehody na silnici tedy hrozí od deváté večer do páté ráno.



Obrázek 32: Rozdělení nehod podle hodin

7.2.3 Výskyt událostí podle dnů

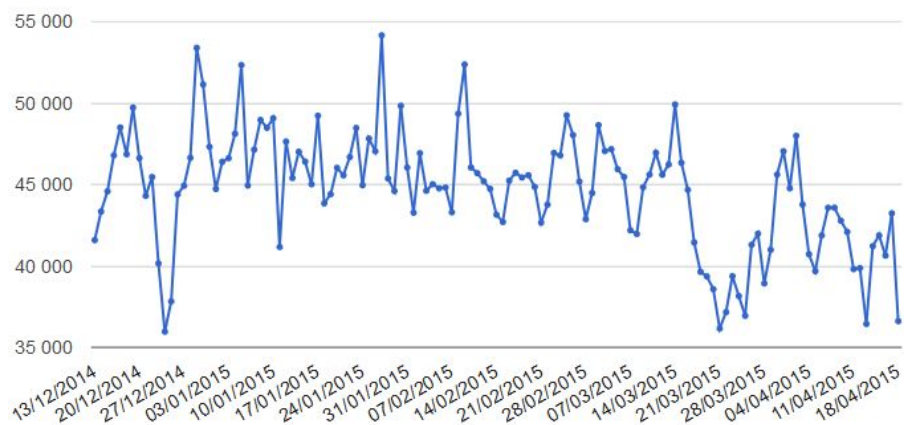
Tento liniový graf zobrazuje počet událostí dle dnů, kdy byly vytvořeny. V grafu lze vypořizovat určité trendy, které se pravidelně opakují. Tomuto trendu se však zcela vymyká hodnota naměřená 31. března 2015. V tento den byl vytvořen skoro dvojnásobek událostí, než je v tento pracovní den obvyklé. Tato hodnota nebyla způsobená žádnou chybou, nýbrž nepřízní počasí.



Obrázek 33: Událostí podle dnů

7.2.4 Výskyt verzí podle dnů

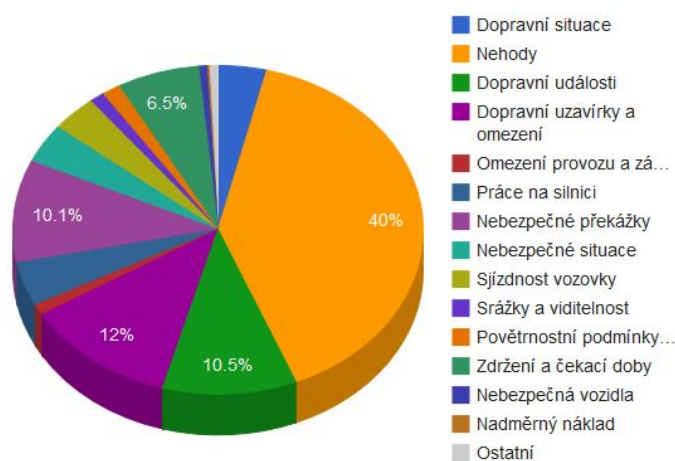
Tento graf zobrazuje počet přijatých verzí k událostem dle dnů. Z grafu lze vypožorovat, že počet denně vytvořených verzí se pohybuje v rozsahu od 35 000 do 55 000.



Obrázek 34: Počet verzí dle dnů

7.2.5 Výskyt událostí podle typu

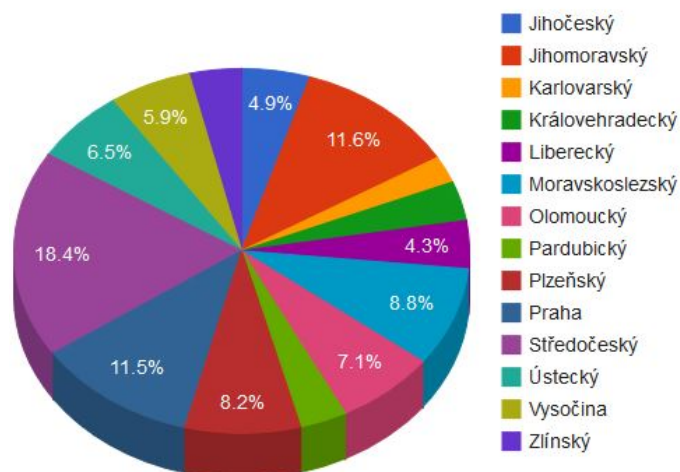
Tento koláčový graf zobrazuje události podle typu. Lze si povšimnout, že skoro polovina přijatých událostí jsou dopravní nehody. V závěsu jsou informace o dopravních uzavírkách. Kolem 10% událostí jsou pak dopravní události a nebezpečné překážky.



Obrázek 35: Události podle typu

7.2.6 Výskyt událostí podle kraje

V tomto grafu je znázorněno rozdělení událostí dle krajů. Nejvíce událostí vzniká ve Středočeském kraji, konkrétně něco přes 18%. Přes 10% událostí je také v Praze a Jihomoravském kraji. Naopak nejnižší výskyt událostí je v Karlovarském a Pardubickém kraji.



Obrázek 36: Události podle kraje

8 Závěr

Tato diplomová práce se zabývá možnostmi zpracování dopravních informací. Při výběru tématu diplomové práce mě zaujalo právě toto téma, jelikož jsem velký fanoušek do GPS navigací. Napadlo mě tedy, že bych se při vypracování této práce mohl přiučit novým zajímavým věcem.

Úvod práce shrnuje, jaké v České republice existují dopravní systémy a na jakém principu fungují. Dále bylo ujasněno, proč je důležité dopravní informace zpracovávat a ukládat. Dále následuje kapitola obsahující informace o systémech, která tato data poskytují a kapitola popisující možnosti lokalizace těchto informací.

V rámci zadání diplomové práce je také požadavek na vytvoření návrhu komplexního modelu, který by umožnil ukládání dat z různých zdrojů. Zdali je však návrh vyhovující, by se ukázalo až při zapojení jednotlivých systémů. Další částí této kapitoly je i návrh modelu, který byl použit při samotné implementaci. Během návrhu bylo dost experimentováno s rychlostí vykonávání jednotlivých dotazů, při kterých jsem si oživil staré a naučil nové techniky.

Součástí práce byla i praktická implementace, která je rozdělena na dvě samostatné části. Každá část je přitom vytvořena odlišnou technologií. První část implementace je zaměřena na získávání, zpracování a následné uložení dat do databáze. Tato část je napsaná v programovacím jazyce C#. Druhá část je vytvořena jako webová aplikace, a jejím hlavním úkolem je vhodně vizualizovat zpracovaná data. Tato část je vytvořena pomocí programovacího jazyka PHP, konkrétně pomocí Nette Frameworku.

Během zpracování obou částí jsem se naučil spoustu nových věcí, které jsem před touto prací ještě nikdy nepoužil. Získal jsem zkušenosti s dolováním dat z XML dokumentů a jejich validací. V druhé části jsem získal některé cenné zkušenosti, především si cením nových zkušeností s psaním Javascriptového kódu, používáním Google Maps API, Google Chart API a dalších. Webová aplikace byla vyvíjena pro co nejvyšší efektivitu, zdali však byl návrh správný, se ukáže až při vyšším počtu návštěvníků. Během vývoje byla aplikace otestována malou skupinou kamarádů, kteří mi poskytli důležitou zpětnou vazbu, pomocí které došlo k vylepšení některých částí.

Poslední kapitola je věnována analýze počtu zasílaných událostí na jednotlivé oblasti České republiky. Z této analýzy lze přehledně získat informace, na které místa v zemi přichází nejvíce dopravních informací. Druhou částí této kapitoly je zobrazení statistických informací o přijatých datech.

Dopravní data pro tuto diplomovou práci jsou získávána ze systému JSDI (viz kapitola 2.1). V průběhu zpracovávání dat nedocházelo k závažnějším problémům, někdy se ovšem stane, že zaslaný XML dokument není validní, nebo neobsahuje povinná data. S tím se však u tak velkého množství dat musí dopředu počítat. Druhý menší problém byl s číslováním verzí (viz kapitola 5.5), ten jsem však vyřešil pomocí vlastního číslování. Rozsah zasílaných dat, bych pro tuto diplomovou práci považoval za vyhovující.

Získávání a zpracovávání dopravních dat je velmi důležité, ať už z pohledu běžného řidiče, tak z pohledu systémů, které s těmito daty pracují. Řidičům dopravní data mohou zajistit bezpečnější a plynulejší cestování. Systémy nad těmito daty mohou modelovat intenzitu dopravy, a do budoucnosti tak můžou pomoci při dalším rozvoji dopravní in-

frastruktury. Do tohoto odvětví tedy určitě má cenu investovat nemalé finance, a to jak na stranu získávání dat pomocí telematických systémů a čidel, tak na stranu zpracování a následného zobrazení.

Do budoucna bych s tímto projektem chtěl dále pokračovat a rozvíjet jej jako informativní pomůcku pro uživatele, které zajímá dění na pozemních komunikacích. Bude proto důležité porozhlédnout se jinde a přidat i některá další data (počasí, snímky kamer, apod.). Při tomto rozšíření se jistě bude hodit i návrh obecného modelu, který je navrhnout v kapitole 4.1. Dále plánuji přidání lokačních tabulek pozemních komunikací (viz kapitola 3.2), které umožní další rozvoj aplikace. V neposlední řadě bych chtěl přidat možnost výpočtu trasy mezi dvěma body. Tento výpočet by bral v potaz i aktuální uzavírky a dopravní situaci. Na zvážení je i přidání API, které by ostatním vývojářům umožnilo získání již zpracovaných dat. Je však otázkou, zdali to licenční podmínky JSDI povolují.

9 Reference

- [1] The Basics of Loop Vehicle Detection. *MarshProducts.com* [online]. 2000 [cit. 2015-04-18]. Dostupné z <http://www.marshproducts.com/pdf/Inductive%20Loop%20Write%20up.pdf>
- [2] Elektronické mýto. *JSDI* [online]. [cit. 2015-04-11]. Dostupné z <http://dopravniinfo.cz/elektronicke-myto>
- [3] Definice předmětu požadovaných služeb plošného monitorovacího systému dopravních toků. *Ministerstvo dopravy České republiky* [online]. [cit. 2015-03-15]. Dostupné z <http://www.mdcrcz.cz/NR/rdonlyres/8A7B37DB-2E52-41E5-B04F-715C999915BF/0/201.pdf>
- [4] Monitoring dopravy. *Oficiální stránky města Zlín* [online]. [cit. 2015-02-19]. Dostupné z <http://www.zlin.eu/monitoring-dopravy-cl-1834.html>
- [5] MANNERING, Fred L, Scott S WASHBURN a Walter P KILARESKEI. *Principles of highway engineering and traffic analysis*. 4th ed. Hoboken, NJ: John Wiley, c2009, xiii, 398 p. ISBN 9780470290750.
- [6] Intenzity dopravy. *České dálnice* [online]. [cit. 2015-04-19]. Dostupné z <http://www.ceskedalnice.cz/odborne-info/intenzity-dopravy>
- [7] Proměnné dopravní značky, zařízení pro provozní informace. *JSDI* [online]. [cit. 2015-03-11]. Dostupné z <http://www.dopravniinfo.cz/promenne-dopravni-znacky-a-zarizeni-pro-provozni-informace>
- [8] Liniové řízení provozu. *JSDI* [online]. [cit. 2015-03-11]. Dostupné z <http://www.dopravniinfo.cz/liniove-rizeni-provozu>
- [9] RDS-TMC obecné informace. *JSDI* [online]. [cit. 2015-04-13]. Dostupné z <http://www.dopravniinfo.cz/obecne-informace3>
- [10] Telematické systémy - obecné informace. *JSDI* [online]. [cit. 2014-04-10]. Dostupné z <http://dopravniinfo.cz/obecne-informace>
- [11] Jednotný systém dopravních informací pro ČR. *JSDI* [online]. [cit. 2014-12-19]. Dostupné z <http://dopravniinfo.cz/jsdi>
- [12] Národní dopravní informační centrum (NDIC). *Ředitelství silnic a dálnic ČR* [online]. [cit. 2015-02-13]. Dostupné z <http://www.rsd.cz/doc/Silnicni-a-dalnicni-sit/Silnicni-databanka/narodni-dopravni-informacni-centrum-ndic>
- [13] Odběr dopravních informací. *Dopravniinfo.cz* [online]. [cit. 2015-02-23]. Dostupné z <http://www.dopravniinfo.cz/odber-dopravnich-informaci>

-
- [14] O centru RODOS. RODOS [online]. [cit. 2015-04-15]. Dostupné z http://centrum-rodos.cz/CZ/o_centru_rodos.aspx
- [15] O systému Czech Traffic. Czechtraffic.cz [online]. [cit. 2015-04-17]. Dostupné z <https://traffic.secar.cz/o-systemu.aspx>
- [16] Analýza dopravní dostupnosti. ČVUT [online]. [cit. 2015-04-15]. Dostupné z <https://www.dopravnidostupnost.cz>
- [17] Pražská doprava. Praha [online]. [cit. 2015-02-15]. Dostupné z <https://kamery.praha.eu>
- [18] Informace o registru UIR-ADR. Ministerstvo práce a sociálních věcí [online]. [cit. 2015-03-03]. Dostupné z <http://forms.mpsv.cz/uir/popis/popis.jsp>
- [19] OKadresy – (nejen) od UIR-ADR k RUIAN. OKadresy.cz [online]. [cit. 2015-03-03]. Dostupné z http://download.arcddata.cz/konf/2012/prezentace/Zindulka_OKsystem.pdf
- [20] CEPK - Dopravní info. JSDI [online]. [cit. 2015-04-01]. Dostupné z <http://portal.dopravniinfo.cz/cepk>
- [21] Global Network. CEDA a ŘSD [online]. [cit. 2015-03-06]. Dostupné z http://www.ceda.cz/files/produktove-letaky/2014/cze/pl_gn_1411.pdf
- [22] RAPANT, Petr. *Geoinformatika a geoinformační technologie*. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, Hornicko-geologická fakulta, Institut geoinformatiky, 2006, xxxv, 463 s. ISBN 80-248-1264-9.
- [23] NATIONAL IMAGERY AND MAPPING AGENCY. *World geodetic system 1984: Its Definition and Relationships with Local Geodetic Systems* [online]. 1. vyd. 2000 [cit. 2015-04-13]. Dostupné z <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>
- [24] GPS - Overview. GPS.gov [online]. [cit. 2015-03-30]. Dostupné z <http://www.gps.gov/systems/gps/>
- [25] Souřadnicové systémy. Západočeská univerzita v Plzni [online]. [cit. 2015-04-19]. Dostupné z <http://gis.zcu.cz/studium/gen1/html/ch02s03.html>
- [26] Síť pozemních komunikací. JSDI [online]. [cit. 2015-04-01]. Dostupné z <http://portal.dopravniinfo.cz/sit-pozemnich-komunikaci>
- [27] Hodnocení hustoty dopravy. Global assistance [online]. [cit. 2015-04-10]. Dostupné z <http://www.globalassistance.cz/?art=431&start=1>
- [28] ALERT-C Codes Project. North American Traffic Working Group - ITS America [online]. [cit. 2015-01-20]. Dostupné z <http://natwg.itsa.wikispaces.net/ALERT-C+Codes+Project>

-
- [29] Distribuce dopravních informací prostřednictvím datového distribučního rozhraní [online]. [cit. 2015-02-19]. Dostupné z http://www.dopravniinfo.cz/public/data/file/Rozhrani_DDR_v3.2.5.pdf
- [30] Odběr dopravních informací z JSDI v praxi. Ing. Jan Vlčínský [online]. [cit. 2015-02-17]. Dostupné z http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2009/sbornik/Lists/Papers/082.pdf
- [31] ConcurrentBag(T) – třída (System.Collections.Concurrent). MSDN [online]. [cit. 2015-03-02]. Dostupné z <https://msdn.microsoft.com/cs-cz/library/dd381779.aspx>
- [32] Rychlý a pohodlný vývoj webových aplikací v PHP. Nette Framework [online]. [cit. 2015-01-16]. Dostupné z <http://www.nette.org>
- [33] AJAX - Tutorial. W3Schools.com [online]. [cit. 2015-01-16]. Dostupné z <http://www.w3schools.com/ajax/>
- [34] JSON: The Fat-Free Alternative to XML [online]. [cit. 2015-03-04]. Dostupné z <http://www.json.org/xml.html>
- [35] Circumference at a Given Latitude. Ask Dr. Math [online]. [cit. 2015-01-19]. Dostupné z <http://mathforum.org/library/drmath/view/54158.html>

A Přílohy

A.1 Struktura DVD s přílohou

Zdrojové kódy a obrázky

- *Docu* – složka s vygenerovanou dokumentací webové části
- *Model* – složka s obrázky modelů
- *Parser* – aplikace určená k získávání a ukládání dat
- *SQL* – složka s SQL skripty
- *Web* – webová aplikace určená k vizualizaci získaných dat

Dokument

- *vym0008.pdf* – tato diplomová práce

A.2 Testovací sestava

Pro testovací účely je pomocí technologie Hyper-V vytvořen virtuální stroj, ve kterém je nainstalován serverový operační systém Windows Server 2012 R2 Datacenter. Aby byl zajištěn určitý výkon tohoto systému, je mu přidělen následující hardware:

- *CPU* – Intel Core i5 4690k (2 jádra)
- *RAM* – 6000 MB
- *HDD* – WD Green 100 GB

Aby však bylo možné s dopravními informacemi pracovat, je potřeba mít nainstalovaný a nakonfigurovaný software. Pro tyto potřeby je použito následující softwarové vybavení:

- *Microsoft SQL Server 2012 Enterprise*,
- *PHP 5.5.22*,
- *Apache 2.4.12*

B Tabulky

B.1 Datový slovník

Sloupec	Datový typ	Null	PK	FK	Popis	IO
administrativeId	INT	NE	NE	ANO	Id z AdministrativeUnit	-
streetCode	INT	NE	NE	NE	Číslo silnice	-

Tabulka 2: Entita AdministrativeStreetUnit

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	NE	Unikátní id	-
msgId	INT	NE	NE	ANO	Id z TrafficData	-
regionCode	INT	NE	NE	NE	Id kraje	-
townShipCode	INT	NE	NE	NE	Id města	-
townDistrictCode	INT	ANO	NE	NE	Id města	-
townCode	INT	ANO	NE	NE	Id města	-

Tabulka 3: Entita AdministrativeUnit

Sloupec	Datový typ	Null	PK	FK	Popis	IO
msgId	INT	NE	NE	ANO	Id z TrafficData	-
roadPart	INT	NE	NE	NE	Id z Číslo silnice	-

Tabulka 4: Entita AffectedRoadPart

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	NE	Unikátní id	-
name	CHAR	NE	NE	NE	Název země	2
area	GEOGRAPHY	NE	NE	NE	Polygon se souřadnicemi země	-

Tabulka 5: Entita Country

Sloupec	Datový typ	Null	PK	FK	Popis	IO
eventCode	SMALLINT	NE	ANO	NE	Složený primární klíč	-
updateClass	TINYINT	NE	ANO	NE	Složený primární klíč	-
descriptioncz	NVARCHAR	NE	NE	NE	Popis typu události	100

Tabulka 6: Entita EventsCode

Sloupec	Datový typ	Null	PK	FK	Popis	IO
versionId	INT	NE	NE	ANO	Id z VersionData	-
eventCode	SMALLINT	NE	NE	ANO	Id z EventsCode	-
updateClass	TINYINT	NE	NE	ANO	Id z UpdateClassCode	-
eviOrder	TINYINT	NE	NE	NE	Pořadí události	-

Tabulka 7: Entita Evi

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	NE	Unikátní Id	-
name	NVARCHAR	NE	NE	NE	Název regionu	50
area	GEOGRAPHY	NE	NE	NE	Polygon se souřadnicemi kraje	-

Tabulka 8: Entita Region

Sloupec	Datový typ	Null	PK	FK	Popis	IO
administrativeId	INT	NE	NE	ANO	Id z AdministrativeUnit	-
roadClass	TINYINT	NE	NE	ANO	Id z RoadClassCode	-
roadNumber	VARCHAR	NE	NE	NE	Název pozemní komunikace	45

Tabulka 9: Entita RoadInfo

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	TINYINT	NE	ANO	NE	Unikátní Id	-
descriptioncz	NVARCHAR	NE	NE	NE	Název regionu	50

Tabulka 10: Entita RoadClassCode

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	NE	Unikátní Id	-
stringId	NVARCHAR	NE	NE	NE	Označení události z JSDI	50

Tabulka 11: Entita StringId

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	NE	Unikátní Id	-
internalPolygonId	INT	NE	NE	NE	Interní označení polygonu	-
numberKm	TINYINT	NE	NE	NE	Velikost okna	-
polygon	GEOGRAPHY	NE	NE	NE	Polygon se souřadnicemi čtverce	-

Tabulka 12: Entita TrafficAnalyticsCache

Sloupec	Datový typ	Null	PK	FK	Popis	IO
cacheId	INT	NE	NE	ANO	Id z TrafficAnalyticsCache	-
trafficId	INT	NE	NE	ANO	Id z TrafficData	-

Tabulka 13: Entita TrafficAnalyticsCacheItems

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	ANO	Id ze StringId	-
type	TINYINT	NE	NE	ANO	Id z TrafficType	-
planned	TINYINT	NE	NE	NE	Plánovaná událost	-
urgency	CHAR	NE	NE	NE	Urgentnost události	1
directionality	TINYINT	NE	NE	NE	Směrovost události	-
diversion	TINYINT	NE	NE	NE	Objízdna trasa	-
coord	GEOGRAPHY	NE	NE	NE	Souřadnice počátku události	-
created	DATETIME	NE	NE	NE	Datum a čas první verze události	-
txpl	NVARCHAR	NE	NE	NE	Textový popis místa	MAX

Tabulka 14: Entita TrafficData

Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	TINYINT	NE	ANO	NE	Unikátní Id	-
name	NVARCHAR	NE	NE	NE	Typ události	5

Tabulka 15: Entita TrafficType

Sloupec	Datový typ	Null	PK	FK	Popis	IO
updateClass	TINYINT	NE	ANO	NE	Unikátní Id	-
descriptioncz	NVARCHAR	NE	NE	NE	Název typu události	65

Tabulka 16: Entita UpdateClassCode

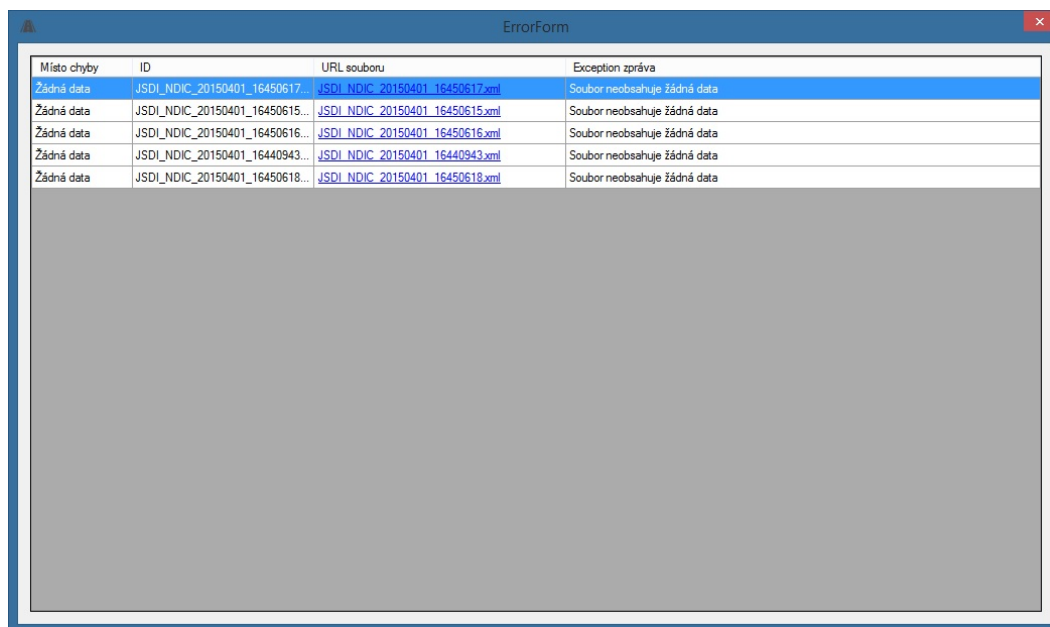
Sloupec	Datový typ	Null	PK	FK	Popis	IO
id	INT	NE	ANO	NE	Unikátní Id	-
msgId	INT	NE	NE	ANO	Id z TrafficData	-
version	INT	NE	NE	NE	Číslo verze události	-
tgen	DATETIME	NE	NE	NE	Datum a čas generování události	-
tsta	DATETIME	NE	NE	NE	Začátek platnosti události	-
tsto	DATETIME	NE	NE	NE	Konec platnosti události	-
otxt	NVARCHAR	NE	NE	NE	Textový popis operátora	MAX

Tabulka 17: Entita VersionData

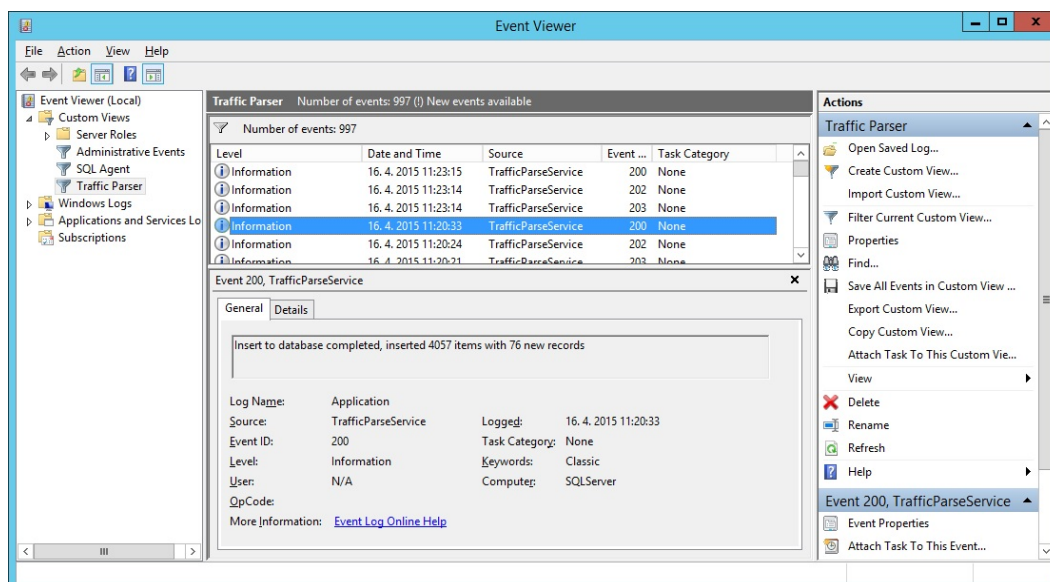
B.2 Lineární schéma závislostí

Název	(rodičovská entita, dětská entita)
FKCacheCacheItems	(TrafficAnalyticsCache, TrafficAnalyticsCacheItems)
FKTrafficDataCacheItems	(TrafficData, TrafficAnalyticsCacheItems)
FKTrafficDataRoadPart	(TrafficData, AffectedRoadPart)
FKTrafficDataVersionData	(TrafficData, VersionData)
FKVersionDataEvi	(VersionData, Evi)
FKEventsCodeEvi	(EventsCode, Evi)
FKUpdateClassCodeEvi	(UpdateClassCode, Evi)
FKTrafficTypeTrafficData	(TrafficType, TrafficData)
FKTrafficDataAdmUnit	(TrafficData, AdministrativeUnit)
FKAdmUnitRoadInfo	(AdministrativeUnit, RoadInfo)
FKAdmUnitAdmStreetUnit	(AdministrativeUnit, AdministrativeStreetUnit)
FKStringIdTrafficData	(StringId, TrafficData)
FKRoadClassCodeRoadInfo	(RoadClassCode, RoadInfo)

C Obrázky



Obrázek 37: Zobrazení okna s výpisem chyb



Obrázek 38: Prohlížeč událostí

D Zdrojové kódy

D.1 Zdrojový kód kontroly mezipaměti

```

private function checkCacheItems($table, $interval = 45)
{
    $filePath = $table . '.neon';
    $fileCacheCheckPath = 'lastCheck' . $table;

    if (! file_exists ($fileCacheCheckPath)) {
        touch($fileCacheCheckPath);
    }

    if (time() - filemtime($fileCacheCheckPath) > $interval) {
        $count = $this->db->table($table)->count('*');

        $data = Nette\Neon\Neon::decode(file_get_contents($filePath));

        if ($data['count'] !== $count) {
            file_put_contents($filePath, Nette\Neon\Neon::encode([
                'count' => $count
            ]));
        }
        touch($fileCacheCheckPath);
    }
}

```

Výpis 5: Ukládání počtu záznamů pro jednotlivé tabulky

D.2 Zdrojový kód výpočtu základu logaritmu

```

function computeLogBase(max) {
    var base = 1.05;
    while (true) {
        var res = Math.log(max) / Math.log(base);
        if (parseInt(res) === 9) {
            return base;
        }
        base += 0.01;
    }
}

```

Výpis 6: Výpočet základu logaritmu